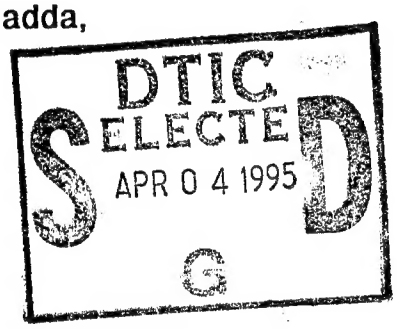
RL-TR-94-218
Final Technical Report
December 1994

INTELLIGENT FINITE ELEMENT SUBMODELING OF MULTICHIP MODULES FOR RELIABILITY ANALYSIS

University of Massachusetts

Ian R. Grosse, Michael Sheehy, Prasanna Katragadda,
and Shankar Raman



APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Materiel Command
Griffiss Air Force Base, New York

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-94-218 has been reviewed and is approved for publication.

APPROVED: Douglas J. Holzhauer

DOUGLAS J. HOLZHAUER
Project Engineer

FOR THE COMMANDER: John J. Bart

JOHN J. BART
Chief Scientist, Reliability Sciences
Electromagnetics & Reliability Directorate

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify RL (ERDS) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1994	Final Jan 93 - Jan 94

4. TITLE AND SUBTITLE
INTELLIGENT FINITE ELEMENT SUBMODELING OF MULTICHIP MODULES FOR RELIABILITY ANALYSIS

5. FUNDING NUMBERS
C - F30602-93-C-0040
PE - 62702F
PR - 2338
TA - 02
WU - PE

6. AUTHOR(S)
Ian R. Grosse, Michael Sheehy, Prasanna Katragadda, and Shankar Raman

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Massachusetts
Department of Mechanical Engineering
Amherst MA 01003

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Rome Laboratory (ERDS)
525 Brooks Rd
Griffiss AFB NY 13441-4505

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
RL-TR-94-218

11. SUPPLEMENTARY NOTES
Rome Laboratory Project Engineer: Douglas J. Holzhauer/ERDS/(315) 330-3476

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Modeling methodologies were developed, implemented, and tested for both rapid thermal finite element analysis of small-scale integrated circuit features in MCMs, and for thermal stress finite element analysis of chip-to-substrate interconnects. A three-step sequential analysis methodology was developed that is initiated with a macroscope thermal analysis of the entire MCM package. The macroscope finite element thermal analysis is then followed by two successive finite element thermal submodels of the hottest die first and then of the hottest die microfeature. In this manner, the thermal analysis process mathematically "zooms" into the hottest IC microfeature without resorting to supercomputer-size finite element models of the MCM. A two-step sequential thermal-stress finite element submodeling analysis procedure was also developed for thermally induced stress analysis of the most highly stressed wirebond or TAB interconnect in an MCM package. For automation purposes, both the IC thermal submodeling and the interconnect elastostatic submodeling methodologies were implemented into an existing blackboard-based, object-oriented MCM software design tool called the Intelligent MCM Analysis (IMCMA).

14. SUBJECT TERMS	15. NUMBER OF PAGES
Finite element analysis, Thermal analysis, Electronic packaging	124
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Acknowledgment

The authors wish to thank Professor Daniel D. Corkill of the Department of Computer Science at the University of Massachusetts for his insights and technical expertise in developing complex software applications. We would like to thank Rome Laboratory engineers Dale Richards, Mark Stoklosa, and Peter Rocci for their cooperation and technical expertise in microelectronic devices and finite element based reliability assessment techniques. Finally, we are very grateful for Doug Holzhauer's excellent management of this research project and for his invaluable technical expertise, guidance, and creative ideas.

Table of Contents

Table of Contents (Cont.)

Appendices

Table of Contents (Cont.)

List of Figures

List of Figures (Cont.)

List of Tables

Abstract

In this research project modeling methodologies were developed, implemented and tested both for rapid thermal finite element analysis of small-scale integrated circuit features in MCMS and for thermal stress finite element analysis of level 1 (chip-to-substrate) wirebond and TAB interconnects. Due to the small size of IC "microfeatures", a three step sequential analysis methodology was developed which is initiated with a macroscopic thermal analysis of the entire MCM package. The macroscopic finite element thermal analysis is then followed by two successive finite element thermal submodels of first the hottest die and then of the hottest die microfeature. In this manner the thermal analysis process mathematically "zooms" into the hottest IC microfeature without resorting to supercomputer-size finite element models of the MCM.

In a similar manner a two step sequential thermal-stress finite element submodeling analysis procedure was developed for thermally induced stress analysis of the most highly stressed wirebond or TAB interconnect in a MCM package. For automation purposes both the IC thermal submodeling and the interconnect elastostatic submodeling methodologies were implemented into an existing blackboard-based, object-oriented MCM software design tool called the Intelligent MCM Analysis (***IMCMA***).

Finally, finite element algorithms for thermal stress analysis involving brick and tetrahedral elements, as well as state-of-the art finite element error estimation algorithms, were implemented into ***FEECA***P, the existing finite element analysis code employed by IMCMA. The finite element algorithms were validated by benchmark comparisons with a commercial finite element code and by mesh convergence studies.

1 Introduction

Multichip modules (MCMs) are high performance microelectronic devices, consisting of several chips mounted and interconnected to a multilayer substrate (Figure 1 [32]). MCMs are currently used in military, aerospace applications and in mainframe computers [1]. Figure 2 illustrates the key features, including interconnect classification, of a basic low cost module containing three dies, which are often referred to as chips. On the top surface of each die is an integrated circuit. The dies may be mechanically mounted to the common circuit base (substrate) with a die attach adhesive and separate first-level electrical interconnections as shown in the figure, or in the case of the flip chip technology a solder bump array is used to both mechanically mount the dies and function as first-level electrical interconnections. Similarly, the MCM package itself may be mechanically mounted to a printed wiring board by an adhesive with separate second-level interconnects for electrical connection, or the package may be both electrically and mechanically to the printed wiring board using a ball grid array of solder bumps.

Figure 3 shows two types of common first level interconnects: tape automated bonds (TAB) and wirebonds [32]. MCMs can fail through the fracture or debonding of a first level interconnect such as the wirebond or TAB (tape automated bonding) bond. Wirebonding is the most common interconnect technology used to make electrical connections between the chips and the substrate. A wirebond is a wire bonded at its ends to the substrate and a chip, by ultrasonic, thermocompression, or thermosonic bonding. The wire material is usually either gold or an aluminum-magnesium alloy. Tape automated bonding can produce a much higher interconnect density than wirebonds. TAB bonds consist of patterned metal, usually copper, attached to polymer tape [2].

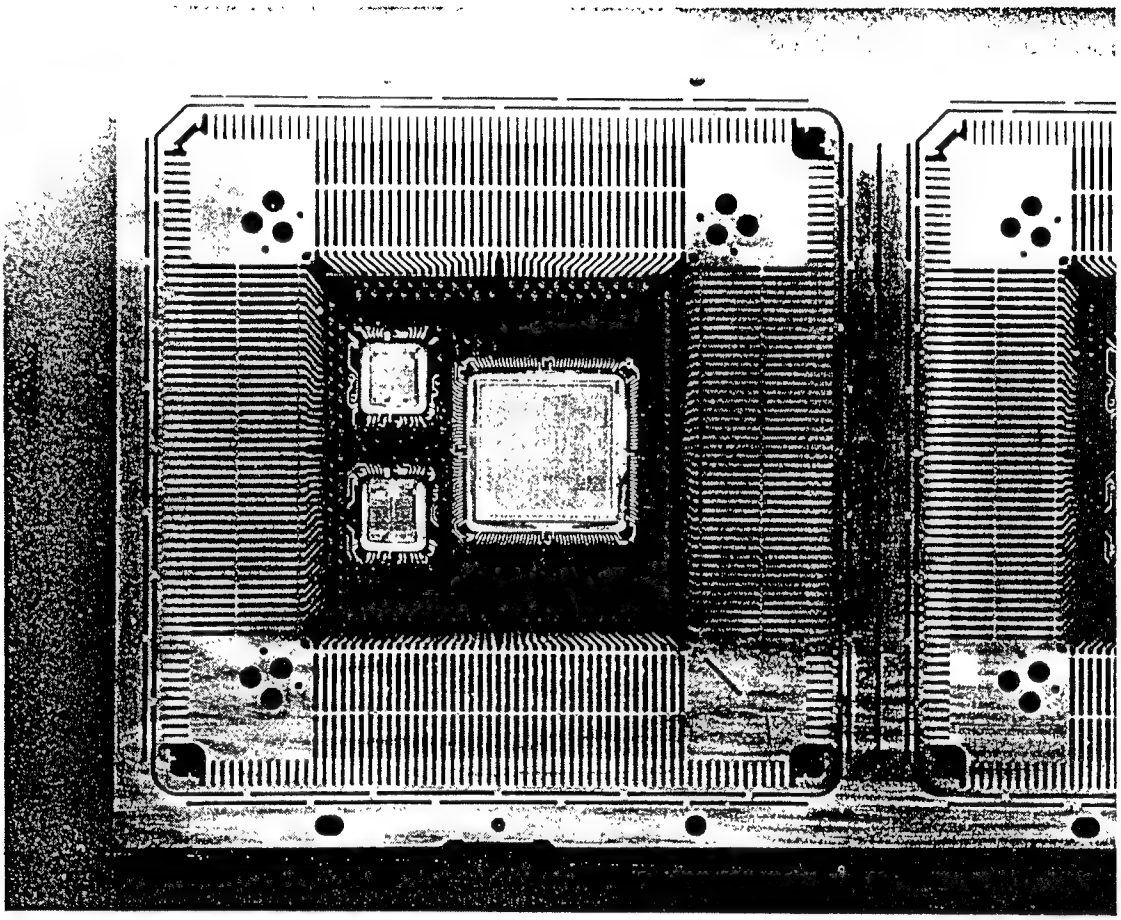


Figure 1: Low Cost MCM (source: *Multichip Module Technologies and Alternatives: The Basics,* edited by D.A. Doane and P.D. Franzon, Van Nostrand Reinhold, NY, 1993, p. 80)

Another form of mechanical failure in MCMs is thermal derating- the overheating of a component. Both temperature and stress values must be examined by the MCM designer to determine an effective mechanical design.

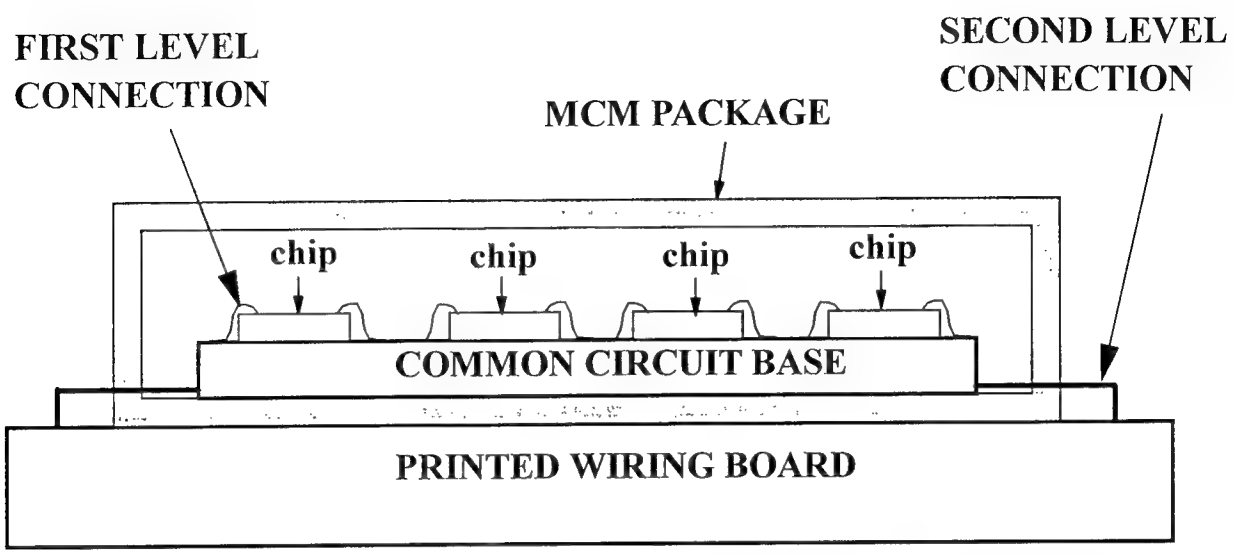


Figure 2: MCM Architecture (schematic), (adopted from *Multichip Module Technologies and Alternatives: The Basics,* edited by D.A. Doane and P.D. Franzon, Van Nostrand Reinhold, New York, 1993, p 5.)

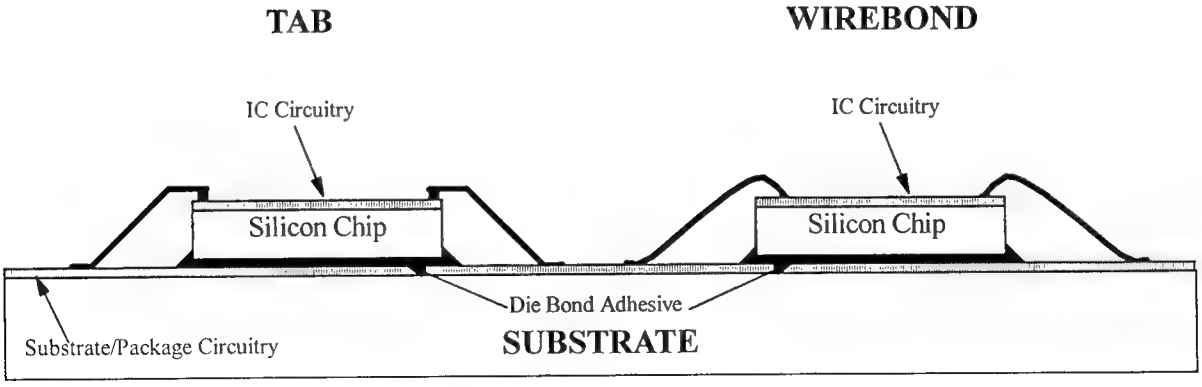


Figure 3: Common types of first level connections: chip to common circuit base (adopted from *Multichip Module Technologies and Alternatives: The Basics,* edited by D.A. Doane and P.D. Franzon, Van Nostrand Reinhold, New York, 1993, p 7.)

MCMs are extremely expensive and time consuming to manufacture, so it is important for design engineers to have tools for determining promising designs early in the design process. Research has shown that evaluation of several design candidates early in the design cycle can reduce the cost and length of the design process and improve the robustness of the MCM [3]. Thus, in the domain of these advanced microelectronic packages, rapid design evaluation via modeling and simulation tools, in lieu of expensive and time-consuming prototyping and testing of various alternative package designs, is critical for selecting the most promising design for subsequent prototyping and reliability testing. Note that given the current limitations in predicting MCM reliability, the purpose of a present-day, design-oriented simulation tool should not be to eliminate MCM prototyping and reliability testing but rather to minimize the number of MCMs prototyped, tested, and re-engineered. It should also be emphasized that in a production environment, especially in the computer industry, time to market is absolutely critical. Therefore, a design-oriented MCM simulation tool must be as automated as possible and provide feedback in minutes or no more than a few hours of real time.

The ***Intelligent Multichip Module Analyzer*** (***IMCMA***) is an automated finite element analysis (FEA) based mechanical design system for MCMs. ***IMCMA*** automatically creates a minimum degree-of-freedom finite element mesh of a MCM from user-defined high-level information and computes the temperature distribution in the large-size MCM features (such as substrate, dies, die attach, package, etc.) through finite element analysis. Thus, a design engineer can use ***IMCMA*** to quickly simulate the macroscopic thermal and mechanical behavior of various MCM package design configurations. While ***IMCMA*** provides the design engineer with important information on the macroscopic thermal mechanical behavior of a MCM design, it does not

provide any data on the thermal or mechanical behavior of microscopic MCM features, such as power dissipating integrated circuit (IC) features and first-level interconnects whose behavior are often critical to the reliability of a MCM package.

2 Objectives

The objectives of this research project are to develop, implement, and test methodologies for predicting both the thermal behavior of power-dissipating integrated circuit features, such as field effect transitions (FET), and the mechanical behavior of MCM first-level interconnects, specifically wirebonds and TABs. It is especially importance that these methodologies be computationally efficient and rapid with sufficient accuracy needed to compare alternative packaging design concepts in terms of relative mechanical MCM device reliability. The methodologies will be instantiated in proof-of-concept software and integrated into ***IMCMA***, a prototype MCM packaging design system. ***IMCMA*** is discussed in more detail in Section 4.

3 Proposed Methodology

The finite element method is a well established, proven algorithmic method for numerically solving the differential equations of equilibrium which govern the behavior of physical systems. However, there are two main obstacles to overcome with regard to the application of the finite element method *as a design tool* to predict the thermal and mechanical behavior of microscopic MCM features such as IC die features and first-level interconnects. First, development of an MCM finite element model with a generic finite element code requires significant human time, finite element modeling expertise, and MCM modeling expertise. Secondly, the sheer number, minute sizes, and the interaction of these microscopic features with the macroscopic MCM behavior

precludes a "brute force" modeling approach in which all macroscopic and microscopic MCM features are geometrically represented in a single, large finite element model. Such a brute force model would consist of tens of thousands of finite elements and nodes, requiring enormous computational resources and time to develop and analyze and thereby obviating the model's usefulness as a MCM design tool.

Clearly, a much more intelligent modeling methodology is needed which can effectively resolve these obstacles. Such an approach lies in the merging of three key, and previously isolated, technologies: 1) high-level, object-oriented data representation, 2) the blackboard-based problem solving paradigm, and 3) finite element submodeling techniques. The first two technologies have been effectively realized in ***IMCMA*** for macroscopic thermal finite element analysis of MCMs. ***IMCMA*** takes advantage of the characteristics of the MCM domain to make modeling simplifications that greatly decrease the modeling and analysis time, while still providing results that are useful to the designer.

Finite element submodeling involves a multistep finite element modeling and analysis process. In the initial step a macroscopic finite element analysis of the entire system is conducted. The system analysis results are used to identify critical regions or features of the model which must now be analyzed in more detail in a second finite element model of only the critical region. The inherent coupling between this critical region and the initial model is handled by imposing boundary conditions on the critical region which have been appropriately obtained from the initial analysis. Essentially, the technique represents a mathematical "zooming" into a critical feature of the model.

An important characteristic of finite element submodeling is that the representation of the critical feature in the initial finite element model can be greatly simplified. Indeed, if the critical feature has negligible effect on the thermal or mechanical behavior of the system model, then the critical feature can even be omitted in the initial system model but then modeled in detail in the subsequent critical-feature model. For example, in many MCM devices, wirebonds and TABs provide relatively insignificant heat transfer and structural stiffness. Thus, wirebonds or TABs can be neglected in an initial thermomechanical finite element model of the entire MCM system. The system results are then used to identify the critical interconnect location in the MCM system. Boundary conditions at this location are extracted from the system analysis results and then imposed on a detailed finite element model of the interconnect to obtain its mechanical behavior.

Submodeling thus involves modeling simplifications based on domain-specific knowledge, i.e. an understanding of the MCM domain and the types of simplifying modeling assumptions which can be made. The system architecture, data representation scheme, and problem solving approach employed in ***IMCMA*** is well suited for encapsulation of domain-specific submodeling simplication knowledge and automation of the entire submodeling process. In section 4 we first present a brief description of the ***IMCMA*** software system, followed by details on thermal and first-level wirebond and TAB interconnect submodeling. In Section 5 benchmark results are presented for both thermal and interconnect submodeling. Verification of the thermal stress finite element analysis algorithms implemented in the finite element solver ***FEECAP*** as part of this research project is contained in Section 5. Conclusions are presented in Section 6. Appendices are attached which provide details on new error analysis algorithms implemented in ***FEECAP*** for

assessing and controlling the discretization error for thermal stress problems, as well as operational details related to thermal and interconnect submodeling.

4 The Intelligent Multichip Module Analyzer (*IMCMA*)

IMCMA is a sophisticated design tool for rapid thermal finite element analysis of MCM package designs. ***IMCMA*** employs is built upon a blackboard system architecture tightly integrated an object-oriented database []-[]. The object oriented database and blackboard system architecture greatly facilitates seamless integration of submodeling software that was developed under this project. Thus, ***IMCMA*** was chosen as the software vehicle for implementing the MCM submodeling methodologies.

4.1 The Object-oriented IMCMA Database

Conventional relational database management systems (DBMSs) are not well suited for applications such as computer-aided design. Relational DBMSs are not capable of supporting the complex relationships between mechanical parts. In a relational DBMS, geometric models are stored in a large number of tables. Explicit links between tables needed to represent interrelations between parts do not exist. The manipulation of these tables then requires large computer programs which have long execution times [4].

Extensive research has been done recently in developing object-oriented databases which are capable of representing complex relationships. ***IMCMA*** uses a database that supports relationships between physical components, and also between components of the finite element model and mesh. We chose an object-oriented database to fulfill this requirement. Storing physical and finite element data in objects provides quick access to data describing any entity. This is due to

the data structure itself, and also to the fact that the data is stored in memory, not on disk. The database groups related information together, and this reduces data access time. If ***IMCMA*** retrieves nodal coordinates and nodal constraints, it does not have to search through different data files to compile this information. Everything is contained in the nodal object. The elimination of disk searches also greatly improves data access time, so ***IMCMA*** is faster than a relational DBMS-based system would be.

In the ***IMCMA*** database, each physical and finite element entity is an object. An object contains information that defines the entity, and also information that defines the relationship of that entity to others. For example, a nodal object would contain not only the nodal coordinates, loads, and constraints, but also its connectivity to other nodes, the elements to which the node belongs, and, if applicable, the geometric surface which contains the node.

The objects are organized into a hierarchy of abstractions or object classes. Figure 4 shows a simplified hierarchy for a three component MCM. The different object classes give the system an ability to focus on specific sets of data. In each step of the analysis, ***IMCMA*** can move up and down the object class hierarchy to an appropriate level of abstraction. Unnecessary details pertaining to other object classes are hidden from the current object class, a property called information hiding. In this way the data is more manageable, and the analysis is faster and more efficient [5]. The combination of abstraction and a blackboard system is particularly effective, because blackboards are adept at moving among multiple levels of abstraction during problem solving [6].

The object-oriented database also allows for an excellent user interface. ***IMCMA*** produces plots of the geometry and the finite element meshes on the screen while running the analysis. The user can choose entities, such as lines, surfaces, nodes, and elements, with the mouse, and all of

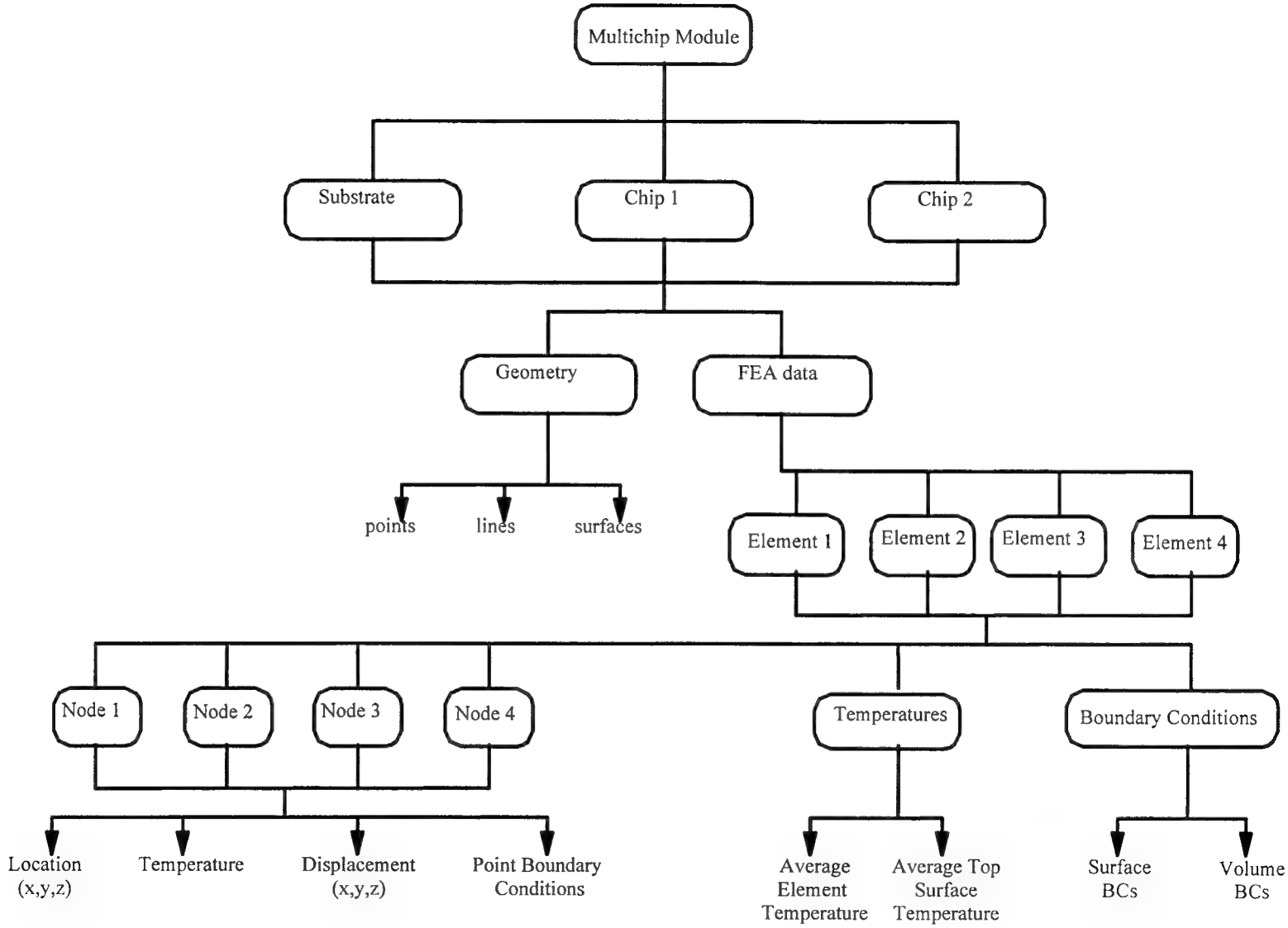


Figure 4: Example Object Hierarchy

the data associated with the entity is displayed.

4.2 Blackboard Systems and Knowledge Sources

IMCMA's object-oriented database exists in the environment of a blackboard system. The blackboard approach is a flexible artificial intelligence (AI) problem-solving technique [7]. We used the GBB™[1] framework, a toolkit for the construction of high-performance blackboard applications, to build ***IMCMA***. GBB supplies the necessary object-oriented blackboard database.

A blackboard system uses three basic components to solve problems [8]:

(1) A global database (blackboard) containing input data, partial solutions, and other data that are in various problem-solving states.

(2) Knowledge sources (KSs) which are independent modules that contain the knowledge needed to solve the problem, and that can be widely diverse in representation and in inference techniques.

(3) A control mechanism, that is separate from the individual KSs and that makes dynamic decisions about which KS is to be executed next.

IMCMA's knowledge sources carry out all modeling and analysis functions, including the definition of geometry, meshing of the model, application of loads and constraints to the finite element model, and finite element analysis of the model.

The current knowledge sources contained in ***IMCMA*** are:

- input-model-ks: defines the device, components, and materials
- adjust-model-ks: adjusts chip geometry for better mapmeshes
- complete-model-ks: creates 2D and 3D lines, points, and surfaces
- generate-mm-regions-ks: generates mapmesh regions
- find-symmetry-ks: calls a CLIPS program for model simplification based on symmetry
- generate-2d-mesh-ks: creates a 2D meshes using FASTQ, a two-dimensional mesh generation tool from Sandia National Laboratory

[1] GBB™ is a product of Blackboard Technology Group, Amherst, MA.

- extrude-component-ks: creates 3D meshes of components by extrusion of 2D meshes, using GEN3D, a mesh extruder from Sandia National Laboratory
- combine-3d-meshes-ks: combines 3D meshes using GJOIN, a mesh joiner from Sandia
- analyze-3d-mesh-ks: performs finite element analysis of model using FEECAP
- model-submodel-transition-ks: identifies component to be submodeled, resets variables and parameters

The blackboard contains all the data needed for the operation of the knowledge sources. ***IMCMA*** stores the data in GBB's object-oriented database, in order to optimize data retrieval for the finite element analysis.

5 Modeling Simplification

One of the major objectives of ***IMCMA*** is to quickly provide designers with information that will help them to make intelligent design decisions. Modeling simplifications are tools used by ***IMCMA*** to construct macroscopic finite element models of MCMs that still provide meaningful results. Finn et al. [9] break down mathematical modeling simplification into two categories, geometric and phenomena simplification.

The four types of geometric simplifications are dimensional reduction, geometric symmetry, feature removal, and domain alteration:

(1) Dimensional reduction: It may be possible to reduce the degree of spatial or temporal analysis. For example, a 3-D model may be reduced to a 2-D model, or a transient analysis may be reduced to a steady-state analysis.

(2) Geometric symmetry: The finite element model can often be reduced by taking advantage of the geometric symmetry. The application of symmetry boundary conditions on the reduced model provides the same physical representation of the system.

(3) Domain alteration: Certain aspects of the physical domain can be changed in order to simplify the analysis.

(4) Feature removal: Features or components which do not greatly effect the macroscopic behavior of the physical system can be eliminated from the finite element model. Correctly done, the removal of features can produce a greatly simplified finite element model while maintaining solution accuracy. For example, the feature may be a possible failure site, and the maximum stress in the feature will need to be calculated.

The use of submodeling provides the ability to remove features at the macroscopic or large-scale level while retaining the ability to perform a detailed analysis on a microfeature or small-scale component of interest. For example, the microfeature may be a possible interconnect failure site, such as the heel/bonding pad interface of a wirebond, and therefore the maximum stress in the wirebond heel needs to be computed. A separate finite element submodel analysis of the wirebond is carried out with the boundary conditions for the submodel nodes interpolated from a previously conducted finite element analysis of the large-scale components that comprise the MCM package. The submodel analysis results provide a stress or temperature distribution of a feature that is likely to be critical. By identifying the critical small-scale region from the macroscopic analysis results and subsequently modeling and analyzing only the microfeature in this region, a huge amount of computational effort is saved. A finite element model of a MCM package with all large-scale and small-scale features included at the onset is highly impractical and may be unnecessary, depending on the domain, since it may be possible to obtain all necessary information through efficient submodeling techniques.

The second category of mathematical modeling simplification is phenomena simplification. Phenomena may be removed or reduced. Phenomenon removal is the omission of an entire phenomenon from an analysis. A decision to complete only a thermal analysis is phenomenon removal, because stress is completely ignored. Phenomenon reduction is the removal of a component of a phenomenon. Ignoring radiation in a heat transfer analysis is an example of phenomenon reduction.

5.1 The Application of Modeling Simplification in IMCMA

Modeling simplifications employed for developing macroscopic MCM models are drawn from knowledge of the MCM domain. At this point, it is important to understand the stage of the design process in which ***IMCMA*** will operate. One of ***IMCMA***'s purposes is to provide mechanical analyses of MCMs whose electrical layout has been completed. The initial design that ***IMCMA*** will analyze has a layout which includes chip and interconnect placement. The die to substrate interconnects are neglected in the intial macroscopic analysis. This modeling simplication is based on the fact that while interconnect materials generally have high thermal conductivity, they have very small cross-sectional area (on the order of 1000 μm^2) which prevents interconnects from being an important heat path. Interconnects are also flexible, so while the displacement of a substrate or chip may stress an interconnect, the displacement of the substrate or chip is not greatly affected by the presence of interconnects. Thus, the removal of the interconnects from the macroscopic model of the MCM will not significantly affect the macroscopic behavior of either temperature or displacement. This feature removal provides a large reduction in computation time. Interconnects are failure sites, however, so they cannot be ignored altogether. ***IMCMA***'s submodeling capability provides the opportunity to model critical interconnects later in

the analysis process, without inclusion of interconnects in the macroscopic model. Submodeling in ***IMCMA***, which will be discussed in the next section, makes the removal of these interconnect features possible.

The macroscopic finite element model can be altered to include the effect of a feature called the die attach without geometrically representing and meshing the die attach feature. The die attach is the layer of adhesive between a chip and the substrate. The layer is very thin compared to the thickness of the chip. This small die attach thickness value would result in high aspect ratio finite elements that model the die attach in the macroscopic. High aspect ratios can introduce large numerical errors into the finite element solution due to the finite precision with which digital computers can represent numbers. To decrease the aspect ratios of finite elements modeling the die attach, a much finer discretization in the xy space of the die attach would be required. This results in a significant increase in the total number of finite elements required to model the entire MCM package and defeats the purpose of developing a simplified efficient macroscopic model to achieve fast, fairly accurate results.

A method which is a form of domain alteration has been developed for this contract to include the effect of the die attach without adding the die attach to the model. In this approach the thermal conductivity of the chip is changed to include the effect of the die attach without physically modeling the die attach. This circumvents the problems of either high aspect ratio elements or too many elements for efficient macroscopic analysis. An equivalent thermal conductivity is calculated by the following formula:

$$K_{eq} = H\frac{k_1 k_2}{k_1 h_2 + k_2 h_1} \qquad (1)$$

where h_1 is the die attach height, h_2 is the chip height, H is the total height $(h_1 + h_2)$, k_1 is the die attach thermal conductivity and k_2 is the chip thermal conductivity.

The critical temperature of an MCM will always be at the top of a chip at the location of the heat producing integrated circuits. By using an equivalent thermal conductivity, accurate temperature values are obtained at the top of the chip with the use of a macroscopic model.

IMCMA also employs another form of domain alteration called xy-adjustment is used to produce simpler mapmeshes. If sides of two chips are nearly collinear, very small elements will be necessary to fill in the space between the two sides. The macroscopic solution will not be affected significantly by a slight change in chip size or location, so the dimensions of the chips can be altered so the sides *are* collinear. These alterations produce a much simpler mapmesh with a significant reduction in total number of finite elements. Therefore, a large amount of computational effort can be saved with almost no loss in accuracy. The engineer has control over the amount of xy-adjustment in chip dimensions that can be introduced into the solution; ***IMCMA*** has an input parameter that controls how much the dimensions will be allowed to change. ***IMCMA*** also has a z-adjustment capability for changing chip thicknesses for better mapmeshes.

IMCMA utilizes several other types of modeling simplification. Code is being developed to identify geometric symmetries of an MCM, and simplifying an MCM finite element model according to these symmetries will reduce the number of nodes by at least a factor of two. Domain alteration is also used through the assignment of a general set of material properties to a multilevel substrate which contains levels of wiring. Other features, such as the lid which covers the MCM package, are left out of the analyses, because they are not critical and have little effect on the physical behavior of the other components.

6 Implementation of Submodeling in *IMCMA*

Two types of submodeling are incorporated into ***IMCMA***: thermal IC die features and wire-bond and TAB interconnects. Thermal die feature submodeling is used to find the on-die temperature distribution of the integrated circuit with the highest surface temperature. Interconnect submodeling is used to identify possible interconnect stress failures in candidate designs. In both cases, submodeling uses the results of the ***IMCMA*** macroscopic thermal or static analyses to get the boundary conditions for the more detailed submodel.

6.1 Thermal Submodeling

Thermal submodeling involves several separate and sequential steps as illustrated in Figure 5 below. The entire process is fully automated in ***IMCMA***. The first step in thermal submodeling is the identification of the hottest chip. This is done by examing the results of the macroscopic MCM thermal analysis. In the macroscopic MCM analysis the heat producing integrated circuit features of each die are approximated by a uniform surface flux applied to the top surface of the die. An average power dissipation value is applied to each surface to ensure that the total power generated is equal to the actual power dissipated on the chip. The use of flux power dissipation allows the mesh to be independent of the geometry of the heat producing regions.

The chip with the highest temperature is then submodeled. The first submodel includes the entire chip and a simplified representation of the heat producing regions. The user can also choose to have the die attach modeled in the first submodel. The chip-base temperature values obtained from the macroscopic MCM thermal analysis are applied as chip submodel boundary conditions. Temperature values at this layer are interpolated, because the submodel has a higher

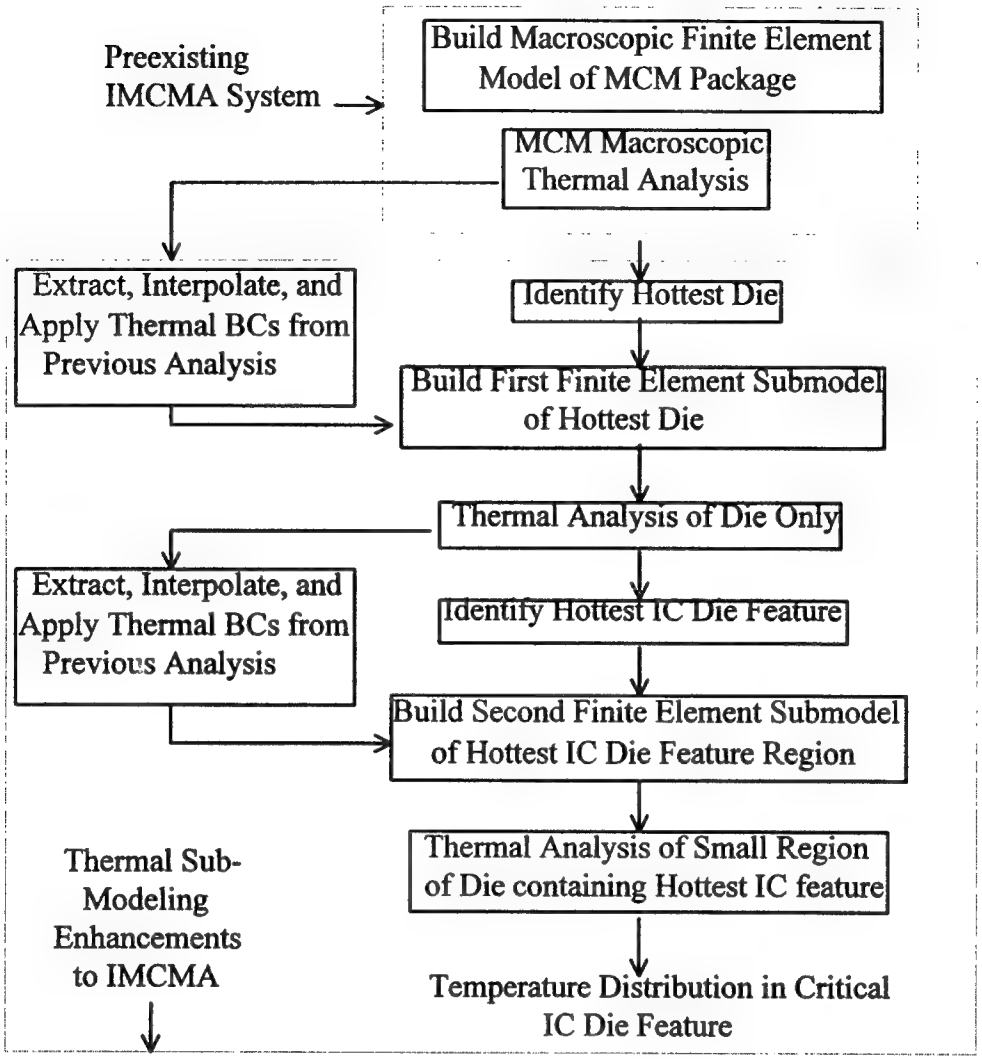


Figure 5: Thermal Submodeling Steps

mesh density than the macroscopic analysis. The interpolation introduces some error, but the analyses are for the comparison of designs, so some accuracy can be sacrificed.

The second submodel contains more exact heat source information and only part of the chip. The amount of the chip that is modeled is dependent on the mesh density of the first submodel, but the maximum thickness is one quarter of the total chip thickness, and the maximum top area is 15 percent of the total chip area. Temperatures are interpolated not only to the bottom submodel

nodes in this case but also to all nodes on the sides of the second submodel. The results of the analysis give the designer an estimate of maximum package temperature. The probability of thermal derating can be assessed from these results, and a better thermal design can be obtained by selecting the design with the lower maximum temperature.

The whole thermal submodeling process is documented on the workstation screen for the designer through graphics and textual descriptions. Geometry, finite element meshes, and thermal analysis results are displayed graphically for the model and submodels. The text takes the user through all knowledge source activations and describes what each knowledge source is completing. The whole submodeling process is built into the knowledge sources described in **Section 4.2 *Blackboard Systems and Knowledge Sources***. The changes that have been made to ***IMCMA*** in order to implement thermal submodeling are detailed in **Appendix D: Implementation of Thermal Submodeling**.

6.1.1 Verification of Thermal Submodeling

A verification of ***IMCMA***'s thermal submodeling capability was completed by comparing ***IMCMA***'s finite element solutions and interpolation values to an ANSYS®[2] analysis of the same MCM. The model used for verification is a simple two chip model. Each chip has nine heat sources mounted on the top surface. Please note that the power dissipation rates are not realistic for an MCM, and that a model of an actual MCM would likely contain a greater quantity of smaller heat sources. The model is only used to validate ***IMCMA***'s thermal submodeling process.

The model consists of two 4x4x0.25 mm silicon chips mounted on a 14x8x1 mm aluminum substrate (Figure 6). A prescribed temperature surface of 20 °C is applied to the bottom surface of

[2] ANSYS® is a registered commercial product of Swanson Analysis Systems, Inc., Houston, PA.

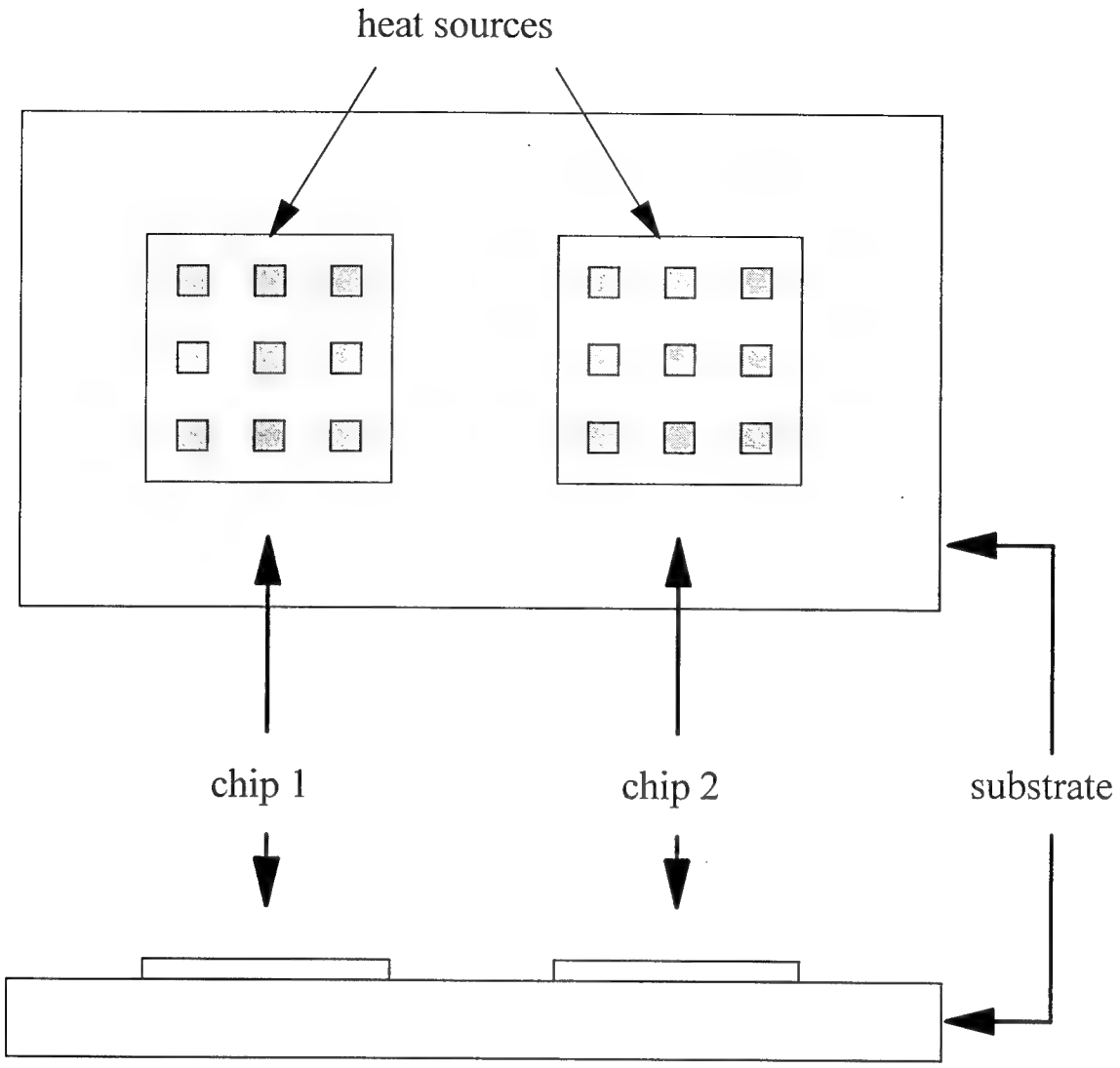


Figure 6: Model for Verification of Thermal Submodeling

the substrate. For the initial macroscopic analysis, uniform power dissipations are applied to the chips. The power dissipation value for each chip is equal to the sum of the power dissipations of the chip heat sources.

The macroscopic analyses were completed on ANSYS and ***IMCMA***. The same mesh is used for each analysis, and this mesh is shown in Figure 7. Figure 7 shows ***IMCMA***'s graphical user interface, which displays model geometry and the finite element mesh. Both the ANSYS and

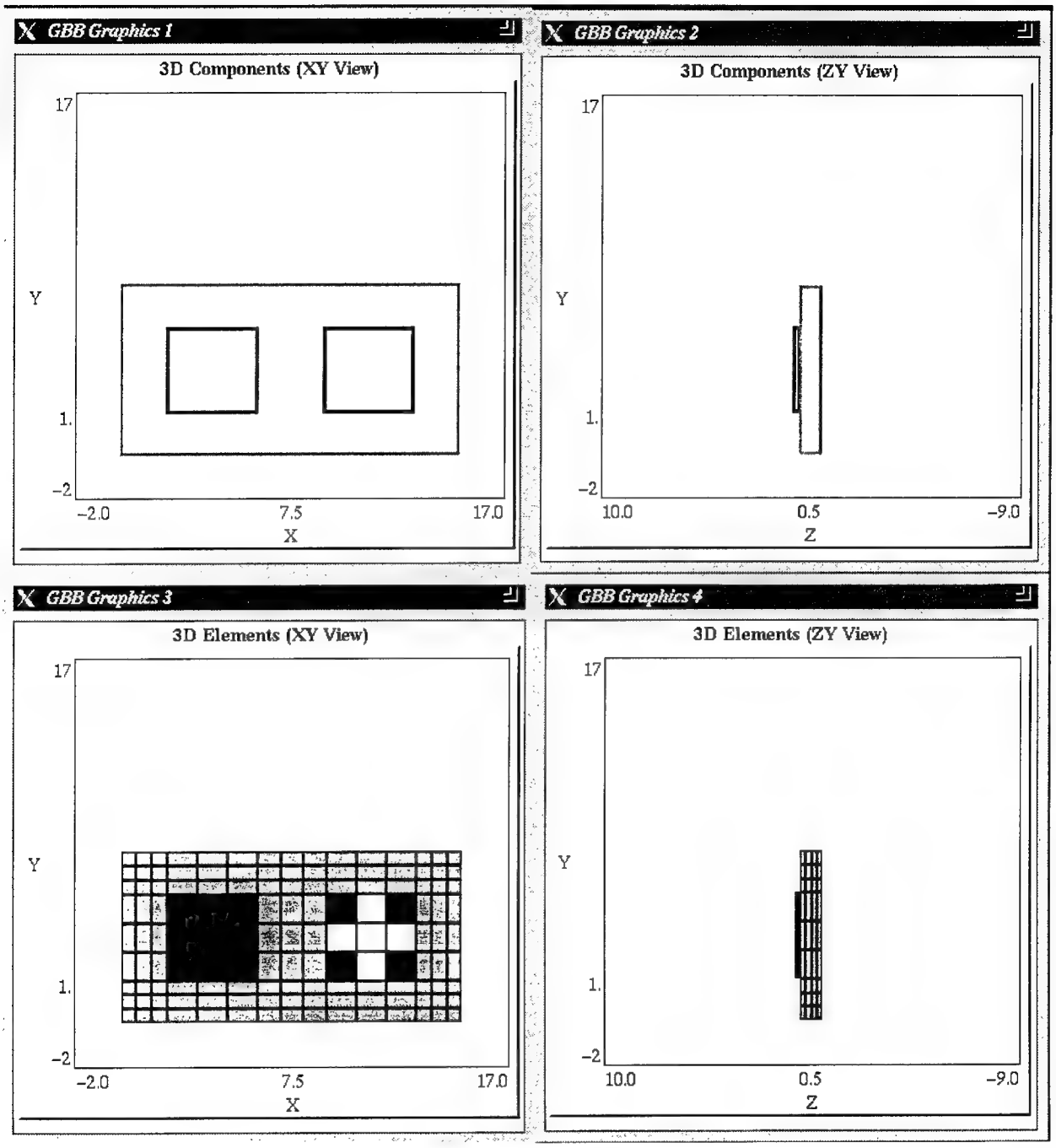


Figure 7: IMCMA Graphics Display for Macroscopic MCM Model

IMCMA analyses produce a maximum temperature of 25.5 degrees on Chip 1. Chip 1 is identified as the hottest chip and is used for the first level submodel.

IMCMA creates the first submodel geometry and applies prescribed temperature values to the bottom of the chip. These values are interpolated from the macroscopic model results. The ANSYS and ***IMCMA*** interpolation values for the verification model were found to agree exactly to two decimal places. The finite element mesh and geometry for the first submodel are shown in Figure 8. After the temperatures are prescribed, ***IMCMA*** examines the heat sources found on the chip and extrapolates the values to the first submodel nodes as point heat sources. ANSYS does not perform this operation, but ***IMCMA***'s point heat source values were examined and found to be correct. The ***IMCMA*** submodel analysis produces a maximum temperature of 112.8 °C at location (4.00, 4.00, 1.25). The ANSYS analysis produces a maximum temperature of 113.3 °C at the same point. ***IMCMA***'s result differs from ANSYS by only 0.51 percent.

The second submodel contains a small region around this point of maximum temperature. The size of the region is equal to the size of 9 first submodel elements (3x3x1). Prescribed temperature values must be applied to the front, back, left, right, and bottom faces. The values are interpolated from the first submodel results. The major difference between the first submodel and the second submodel is the representation of the heat sources. In the first submodel the heat sources are applied to the nodes as point heat sources. In the second submodel the actual geometry of any heat source within the critical region is included. The physical heat source is meshed along with the critical part of the chip, and a heat generation rate is applied to the volume of the heat source.

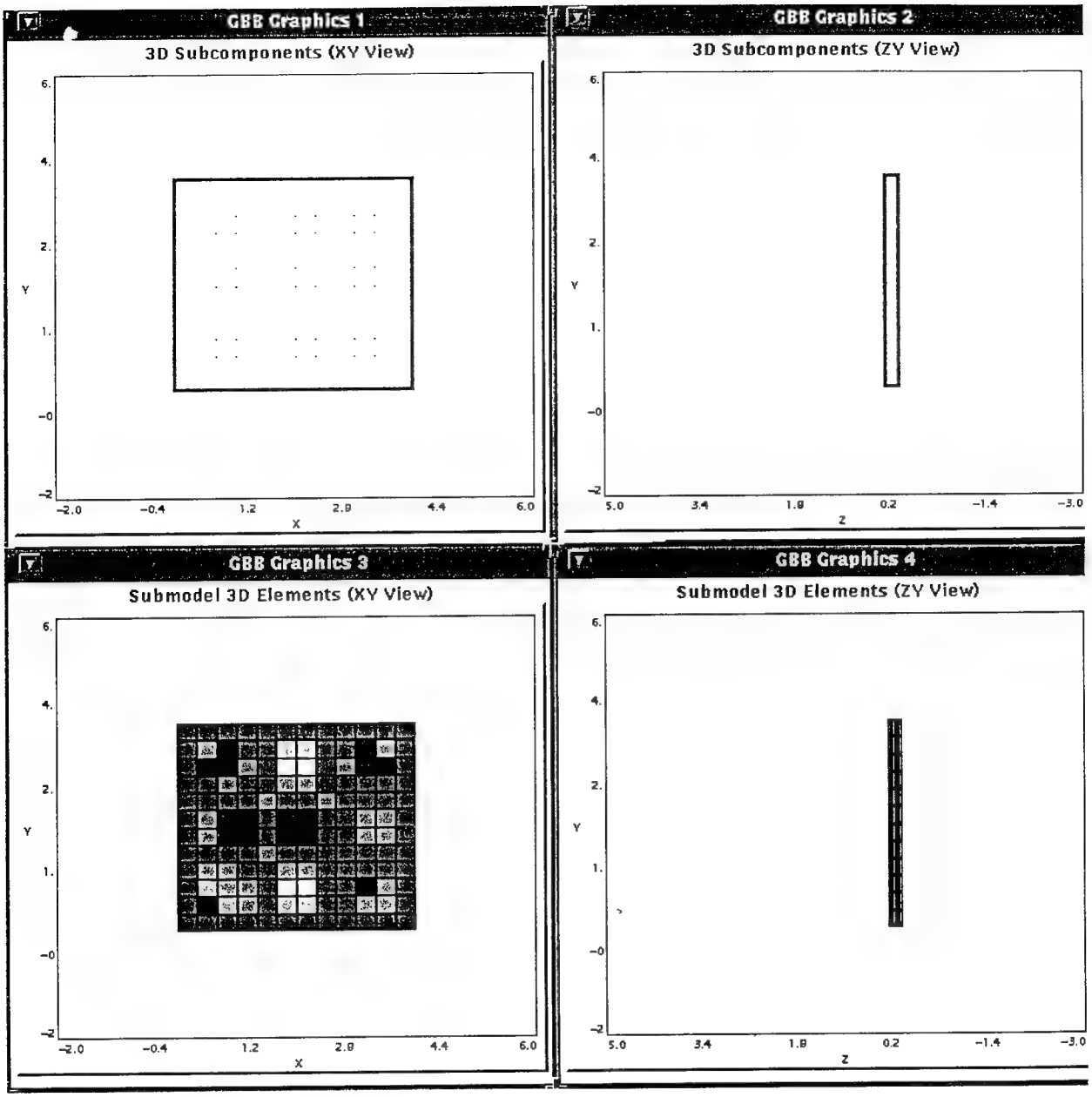


Figure 8: IMCMA Graphics Display for First Submodel

The ANSYS model used for verification had a coarser mesh than the ***IMCMA*** model, but the results were still very close. Figure 9 shows the ***IMCMA*** second submodel and mesh. The maximum temperature in the ANSYS model is 59.9 at location (2.67, 4.00, 1.30). ***IMCMA***'s results compare well, with a value of 61.5 at location (2.75, 4.00, 1.25). The difference in temperature values is likely the result of a slight difference in location and the difference in mesh density.

It should be noted that the first submodel analysis has a maximum temperature nearly twice as high as the maximum temperature of the second submodel analysis. This is due to the fact that point heat sources are used to model IC power dissipation in the first submodel, while the physical IC power dissipating component is modeled in the second submodel as a three-dimensional solid with volumetric heating. High temperatures will be produced at the points where the heat sources are concentrated. These high temperature values can still be used to correctly identify the critical region of the submodel, through the additional consideration of the size of the physical heat sources. The maximum nodal temperature on an element which contains heat sources is divided by the area of the heat sources contained in that element. The element which has the highest temperature to area ratio is identified by ***IMCMA*** as the center of the critical region.

6.2 Interconnect Submodeling

The second type of submodeling methodology developed as part of this contract is first level chip-to-substrate connections. The different types of first level connections in MCMS are [cite RL-TR 92-95].

- wirebond
- tape automated bonding (TAB)
- controlled collapse chip connection (C4)
- polymer overlay/direct metallization on die (high density interconnection)

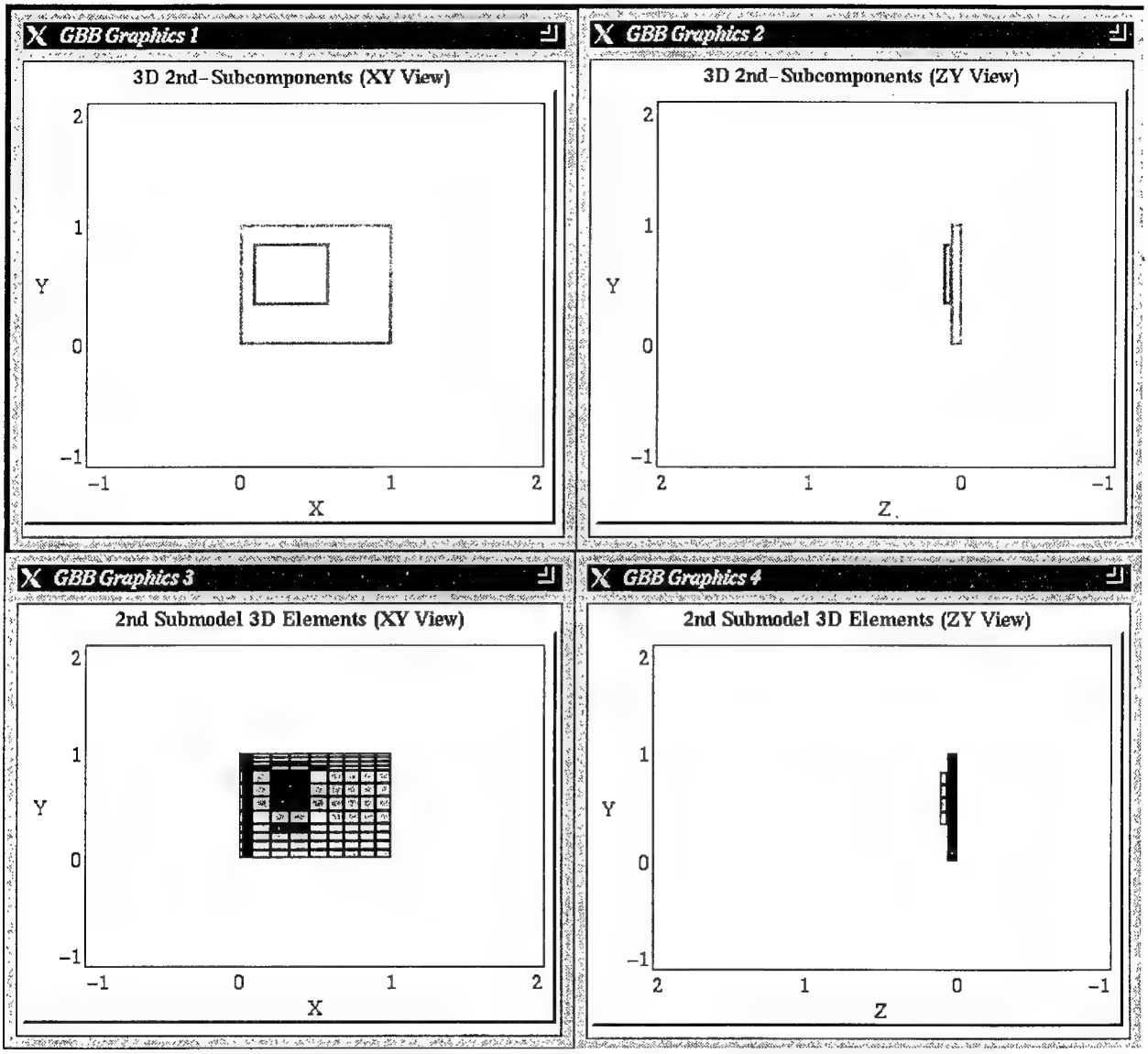


Figure 9: IMCMA Graphics Display for Second Submodel

- laser pantography

The scope of this contract is limited to wirebonds and TABs which are established first-level interconnection technologies. The C4 configuration also called "flip-chip" or "solder-bump" bonding technology provides the highest possible number of I/O's and is also suitable where low inductance is an important consideration in the design. The main drawback in this process is the requirement for a full area array of first level connection bonding pads on the chip surface. The vast majority of chip makers design and manufacture chips with bonding pads on the chip surface peripheral. IBM, the developer of the C4 technology, is the key exception. Polymer overlay/direct metallization was developed by General Electric and has now been licensed to Texas Instruments. Laser pantography was developed by Lawrence Livermore National Laboratory. This technology has now been transferred to a DoD contractor.

The most common bond type used is the wirebond. Figure 10 shows some details of a wirebond from the modeling viewpoint. Intensive research has been conducted in the area of wirebonding to determine optimum contours of the wirebond and the stress distribution in them. One of the most important reasons for the popularity of this type of bonding is the low cost involved in the process. The automated wirebonding procedures have lead to fast and efficient bonding connections as compared to the manual bonding that used to be previously employed. Some advantages of wirebonding are

- It is possible to use various metallizations.
- Adaptibility to wider substrate tolerances.
- Amenability to visual inspection.
- Satisfactory operation characteristics and reliability.

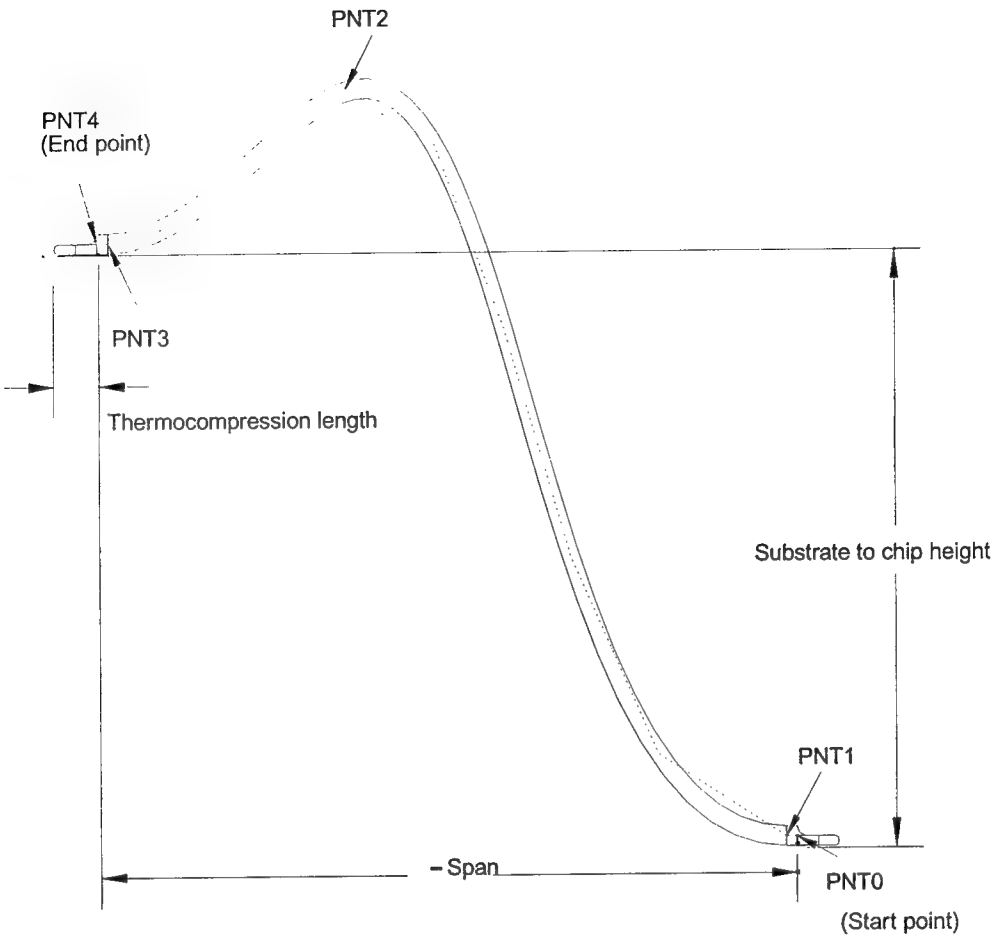


Figure 10 : Wirebond Pro/Engineer Solid Modeling Parameters

The TAB technology, however, seems to be taking over the wirebonding technology mainly because of the number of connections that can be made to a chip. Figure 11 shows the lay-out of TAB tape of the inner lead bond and outer lead bond. The inner lead bonding consists of at-taching the unsupported conductors on the tape to the die. This is normally done using the process of "gang bonding" i.e simultaneous bonding of the leads to the bond pads. The usual bonding methodology used is the thermocompression (T/C) bonding. Thermocompression bonding is dis-cussed later in this section. Once the inner lead bonding is completed, the chip with the leads is placed on the substrate and the leads are thermosonically or ultrasonically bonded to the the

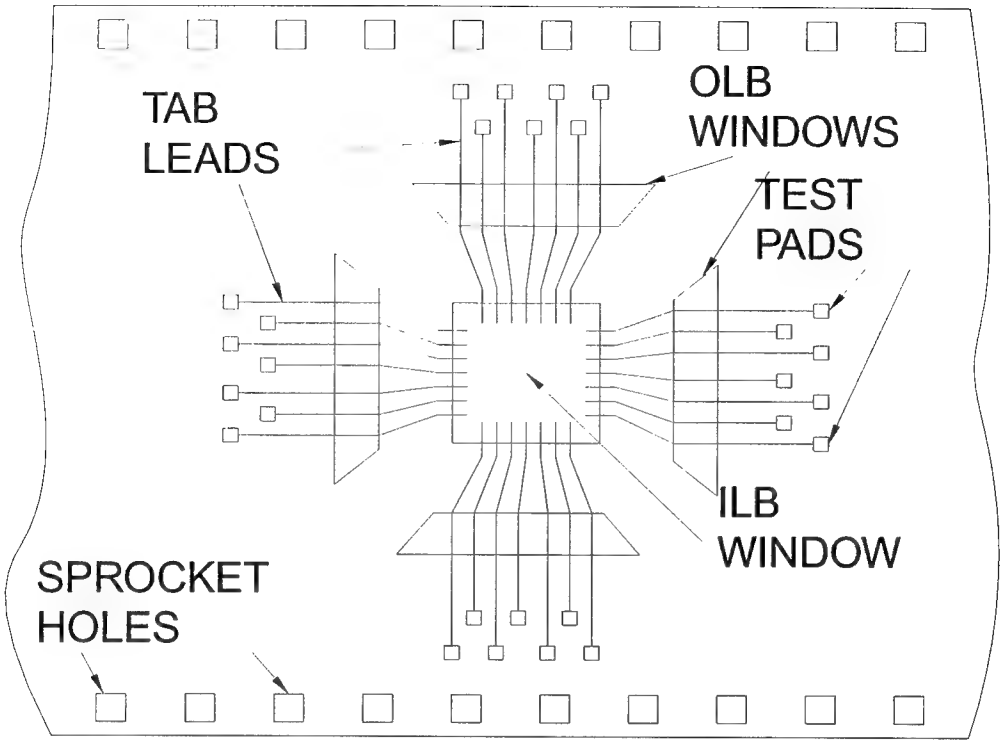


Figure 11: Features of TAB Tape

substrate. These two bonding techniques are also discussed in brief later. Some advantages of the TAB over wirebond include the following [cite IPC-MC-790 and Micro. Pack. HB, p. 434].

- improved electrical performance at high frequencies
- ability to test at AC speed and temperature, and burn in the die prior to final assembly
- improved heat dissipation
- lower bond profiles and improved component reliability
- shorter interconnect lead lengths, resulting in higher speed performance

This report discusses the modeling and analysis techniques of the most commonly used chip-level bonding technologies - wirebond and TAB.

6.2.1 Bonding Methodologies and Bond Types

The type of manufacturing technique to be employed while bonding the lead to the chip or substrate affects the strength of the bond and its operational characteristics. Wirebonds are normally bonded using three different bonding techniques - ultrasonic (U/S) bonding, thermocompression (T/C) bonding and thermosonic (T/S) bonding. In ultrasonic bonding, the wire is clamped onto the bond pad and the bonding is achieved by applying ultrsonic energy normally at ambient temperature. A metallurgical "cold weld" between the wire and the bond pad results. Thermosonic bonding is similar to U/S bonding, the only difference being the slightly elevated temperatures of the component used in the process. Ball bonding and wedge bonding can be carried out using U/S and T/S methods. Figure 12 shows a schematic of the ball and wedge bonds. Thermocompression (T/C) bonding does not use ultrasonic energy. The bonding is achieved by forcing the wire on the bond pad and using thermal energy to raise the temperature (~ 300°C) at the bond/surface interface. T/C bonding is also used for ball and wedge bonding. However due to the long bonding times and the high temperatures required in the process, it is becoming less

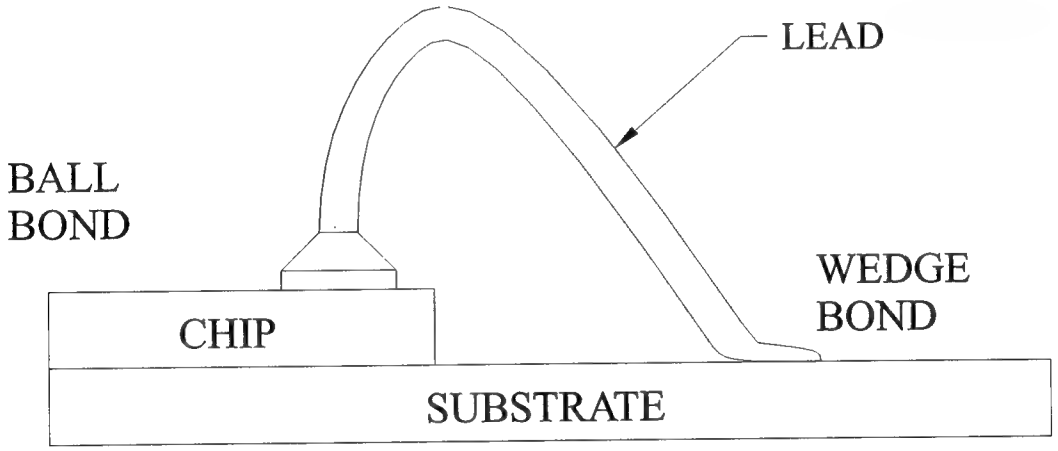


Figure 12: Bonding Types- Ball Bond and Wedge Bond

popular than the other two methods. The process of determining the critical interconnect, its modeling and analysis are discussed in the following sectiion.

6.2.2 Identification, Modeling and Analysis of the Critical Interconnect

Interconnect submodeling interfaces the commercial solid modeler/mesh generator Pro/ENGINEER with ***IMCMA***. Interconnects are geometrically complex, unlike rectilinear MCM components, and cannot be modeled and meshed automatically by ***IMCMA***. Parametric models of interconnects are created in Pro/ENGINEER, and ***IMCMA*** provides values for the necessary parameters, fires Pro/ENGINEER to mesh the model, imports the mesh data back into the ***IMCMA*** database, and performs the finite element analysis. Finally the results are also read back into ***IMCMA***.

The first step in interconnect submodeling is the identification of a critical interconnect. Research data has shown that critical wirebonds and TAB bonds are found at the corners of the chips. Displacements and temperatures are found at these corner locations from the macroscopic thermal/static analysis of the MCM. Displacement causes stress in the material by bending the interconnect, and temperature causes stress through the mismatch in coefficient of thermal expansion between the interconnect and the chip or substrate. Relative importance of temperature and displacement would have to be determined through an extensive study, so until this can be done, a simple methodology is used to identify the critical interconnect. The interconnect at a corner location with the highest relative displacement between its ends is taken to be critical. ***IMCMA*** checks the relative displacements at all corner locations, and finds the maximum.

The interconnect model is interactively created by the user in Pro/Engineer and meshed. The displacements at both ends of the interconnect are applied as boundary conditions to the

Pro/engineer finite element mesh. ***IMCMA*** performs a finite element analysis of the interconnect, and the designer has access to all results, which are stored in the database.

6.2.2.1 Identification of the Critical Interconnect

In order to zoom into the critical regions of the MCM and to extend the analysis to a microscopic level, it is necessary to identify areas of the MCM that are prone to failure. Typically, the solution to this is submodeling. Submodeling involves determining the features that may be critical to the design, modeling them and carrying out the analysis. Boundary conditions on the submodel are specified using information from the macroscopic analysis.

The interconnect is one such component of the MCM which tends to fail and hence needs to be submodeled. Each MCM may contain anywhere between 100 to 500 interconnects and analyzing all of them would take up a large amount of computation time. This necessitates the identification of the critical interconnect given a set of interconnects.

In order to identify the critical interconnect from a user input of interconnects a critical interconnect knowledge source was developed. This knowledge source identifies the critical interconnect based on the largest relative displacement between the start and end point of the interconnect and stores the attributes of the critical interconnect on the blackboard. The basic concept for the identification of the critical interconnect is discussed below.

Initially a macroscopic mesh analysis is performed with only the substrate and chip configuration i.e. excluding the interconnects. The nodal displacement and nodal temperature information of the macroscopic mesh is then present on the blackboard and can be retrieved when required. The interconnect geometry is defined by its start and end points. These points are located with respect to the macroscopic mesh configuration. The elements containing the start and end points of

the interconnect are identified, and the displacements are interpolated from the nodes of these elements to the start and end points. The relative displacement is then computed for the interconnect. This procedure is carried out for each "chip corner interconnect" and the interconnect with the largest relative displacement is determined. This interconnect is stored on the blackboard for further analysis ("zooming" onto the critical interconnect). All other interconncts are deleted from the blackboard database.

A brief description of each module or knowledge source which was developed/modified for the identification of the critical interconnect is contained in Appendix F, Determination of Critical Interconnect .

6.2.2.2 Modeling Of the Interconnect

As discussed earlier, interconnects are sensitive components of an MCM. Size and construction are important factors in interconnect failure, so an authentic model and an accurate analysis procedure are necessary. The complicated construction features of interconnects introduce modeling complexities, which if neglected may lead to errors in the final results.

The two classes of interconnects that have been our main area of concentration in this report are *wirebonds* and *TAB* (tape automated bond) bonds. While the latter does not pose too many modeling challenges, the contour of the wirebond demands the use of a variety of modeling skills. A basic assumption of a *skilled modeler* is made while designing and executing the system. The modeling of the interconnect is carried out using the parametric modeler in Pro/ENGINEER.

CONSTRUCTIONAL AND FUNCTIONAL DETAILS OF THE INTERCONNECT:

The interconnect consists of a gold or aluminum alloy lead, typically about 0.005 inches in diameter (for a wirebond) or width (for a TAB). The entire span of the interconnect is of uniform

cross-section except the ends are flattened out or given a ball shape to facilitate bonding. The flat end region is known as the thermocompression. It is the region on the interconnect which establishes the electrical connection between the chip and the substrate and also serves as a reinforcement to the chip. The interconnect has a sudden change in the cross-sectional area in the region of the thermocompression. This area, referred to as the *bond heel,* is the most critical region of the interconnect due to a concentration of stresses. Some previous tests to determine fatigue life of interconnects have shown failure at the bent arc (the topmost region of the interconnect) but this has been mainly due to surface asperities, manufacturing flaws and material defects. Failure normally occurs at the heel of the interconnect.

MODELING CONSIDERATIONS AND MESHING:

The modeling methodology consists of interactive use of Pro/ENGINEER. The critical interconnect needs to be modeled and analyzed. The user is initially queryed by ***IMCMA*** for the material of the interconnect. This material and its properties must be defined in the example file. Depending on the number of chips in the model, the four corner locations of each chip are considered to be the end points of the interconnect. User input for the span of the interconnect is taken and the start points are determined. The assumption made here is that the material and span of all four interconnects of a chip are the same.

Once the interconnect is identified, a Pro/ENGINEER window is started up. The start point indicates a point on the substrate while the end point is located on the chip. Figure 10 on page 27 shows the start and end points of the interconnect and the thermocompression region. To start the modeling procedure, the user needs to read in a file of points (*points.dat*) that define the interconnect contour. This file is created by ***IMCMA*** while determining the location of the critical

interconnect. It consists of five points in case of a wirebond (the start-point, end-point and three other intermediate points) and four points, the start and end-points and two other intermediate points in case of a TAB bond. These points are read in as datum points and are reference locations for the interconnect model. The start point is always at the coordinates (1,1,1) and the end point is displaced relatively from the start point depending on the span of the interconnect. These start and end point coordinates may not match those of the critical interconnect. Only the relative x, y and z offsets are the same. This change is done to simplify the task of modeling and also the analysis procedures. From here on the modeling is carried out by a skilled modeler. There may be a large number of methods that may be employed in modeling the interconnect. The choice of this method is left to the user. Some methods are suggested in Appendix G.

Once the model has been created, the user meshes the model using the *tet-mesh* facility in the FEM module of Pro/ENGINEER. The default value of global element length may *not* be used as the resulting meshes have a large percentage of distorted elements. Once the automatic mesh has been created, the user outputs the model through Pro/Engineer to an ANSYS input file (log file) using the interfacing capability of Pro/ENGINEER. The file must be named *bond.ans.* The element type used is a linear 4 noded tetrahedral element. At this point there are no boundary conditions applied to the model. The model may be stored by the user and he/she may exit Pro/ENGINEER.

6.2.2.3 Reading Mesh Data

Once the ANSYS input file of the interconnect mesh has been created, the mesh data is read into ***IMCMA*** from the *bond.ans* file. All nodes and elements in the model are read in and created as objects on the blackboard. All nodes of the interconnect are created as instances of the unit

class *interconnect-3d-node* while the elements are created as instances of the unit class *interconnect-3d-element* . Each instance of the node contains information on the x-y-z coordinates and is linked to the respective elements through a link slot. The name assigned to the node is its number. The instances of the elements hold information on the nodal instances that make up the element and the nodal connectivity represented as a node-list. The mesh data is linked together to enhance information retrieval using different GBB commands.

6.2.2.4 Analyzing The Model

The blackboard contains the following information at this stage of the modeling process.

- Information about the nodes and elements.
- Information about the material of the interconnect and its properties which are accessed from the material library .
- Start and end displacements of the critical interconnect obtained from the previous knowledge sources.

The start and end displacements of the critical interconnect are considered to be uniform over all the nodes that are in contact with the substrate and the chip respectively. These are the only constraints applied to the model during the analysis.

The nodes that lie at the start and end are determined using their z-coordinates and the length (x-distance). In case of a wirebond, this x-distance is the thermocompression length. The nodes on the interconnect along the surface of contact between the interconnect and the substrate/chip are assigned the start displacements or end displacements depending on whether the node lies at the start or end of the interconnect.

The mesh information is now written into the INPUT.DAT file. This file is the input for *IMCMA*'s finite element analysis code FEECAP. The data written out to this file includes nodal data, elemental data, material data, loads and other boundary conditions for the analysis. Additionally it contains flags for the analysis type and the type of element being used. In this case the analysis is always a static analysis (flag IATYP = 2) and the element is always a 4 noded tetrahedral element (NELTYP = 3). The analysis is carried out and the mesh results, namely nodal displacements and stresses, are calculated.

The analysis results are written out to the file RESULTS.OUT. In addition the displacements and stresses are written to temporary files DISP.OUT and STR.OUT to facilitate reading the results into *IMCMA*.

Reading the analysis results constitutes assigning the nodal displacements (x, y, z) and stresses (x, y, z, xy, yz, zx, and Von-Mises stress) to the corresponding accessors of the slots in the unit class interconnect-3d-node. The displacements and stresses are stored as a list of three and seven elements respectively. This completes the entire run of the interconnect submodeling module in *IMCMA*.

7 Thermal Stress Formulation and Benchmarking:

7.1 The Eight Noded 3D Linear Brick Element

The eight noded linear brick element is probably the most widely used element type in a variety of design engineering problems that require an accurate 3D finite element meshing and analysis. Within *IMCMA*, since the domain is restricted to MCM package design, the brick element has been chosen for macroscopic finite element analysis of the MCM. The formulation and

system equations for the eight noded thermal brick element were included in the June 1993 technical report [31]. The following section lists some of the element formulation equations for the eight noded brick element for static analysis. A more detailed description of a generic 3-D element formulation for static analysis can be found in Appendix E.

7.1.1 Element Formulation

The formulation of the static element involves the applied static loads and initial strains due to applied thermal loads. The following are the element equations for the eight noded isoparametric brick element. The equilibrium equations for static analysis case for element *e* are given by:

$$[K^e]\{d^e\} = \{r^e_{\bar{\tau}}\} + \{r^e_f\} + \{r^e_{\varepsilon_o}\} \tag{2}$$

where

$$[K^e] = \int_{\Omega^e} [B^e]^T [E^e][B^e] d\Omega \tag{3}$$

$$\{r^e_{\bar{\tau}}\} = \oint_{\Gamma^e} [N^e]^T \{\bar{\tau}^e\}\, d\Gamma \tag{4}$$

$$\{r^e_f\} = \int_{\Omega^e} [N^e]^T \{f\} d\Omega \tag{5}$$

$$\{r^e_{\varepsilon_o}\} = \int_{\Omega^e} [B^e]^T [E^e] \{\varepsilon^e_o\} d\Omega \cdot \tag{6}$$

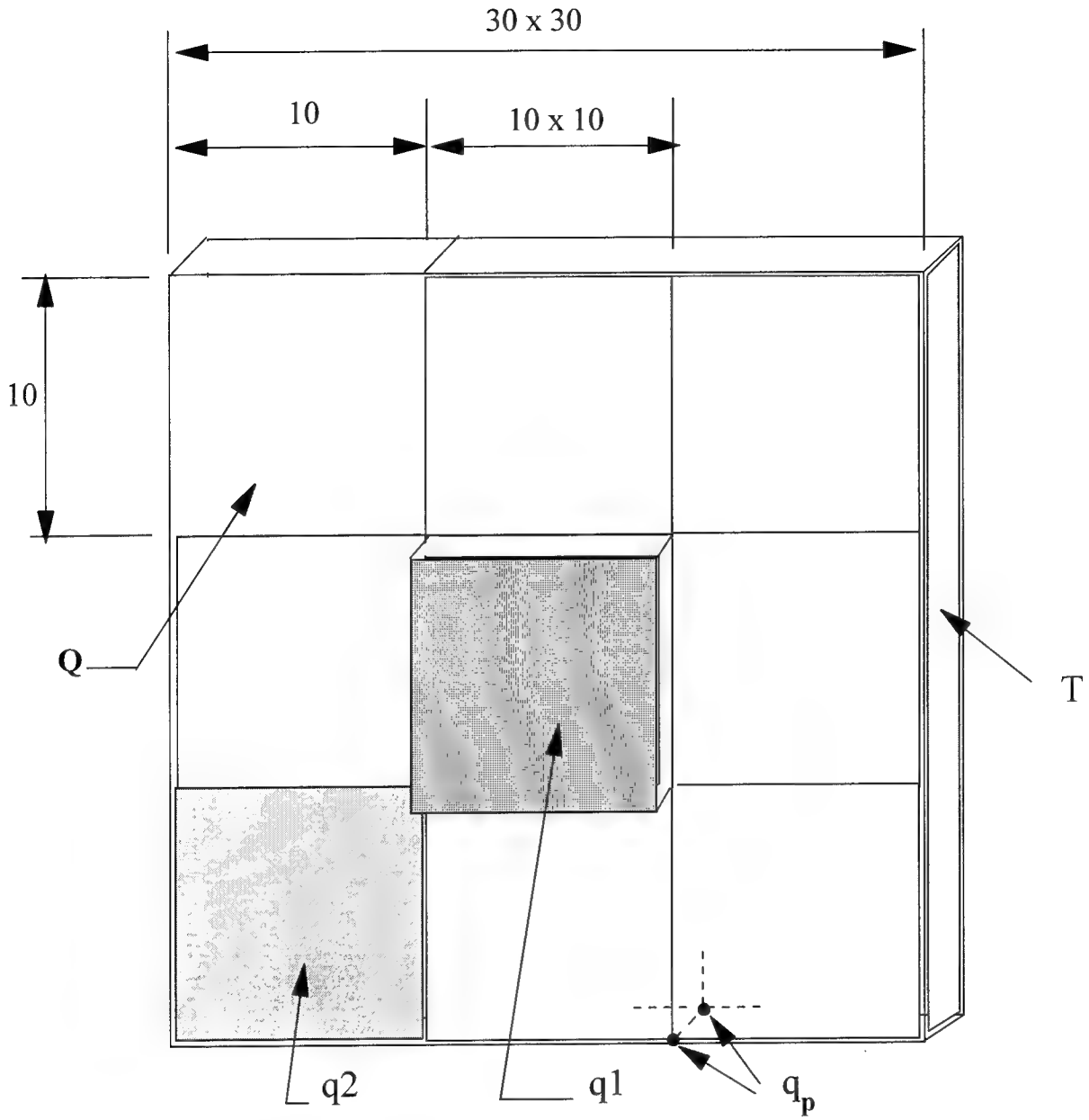
and Ω^e and Γ^e donate the element volume and element surface area, respectively. For static analysis the matrix $[K^e]$ is the element stiffness matrix, $\{d^e\}$ is the vector of element nodal displacements, $[E^e]$ is the material constitutive matrix of the element, $[B^e]$ is the element strain displacement matrix, and $\{r^e_{\bar{\tau}}\}, \{r^e_f\}$, and $\{r^e_{\varepsilon_0}\}$ are the element load terms due to applied surface tractions $\{\bar{\tau}\}$, applied body loads $\{f\}$, and initial strains $\{\varepsilon_0\}$, respectively. The elemental initial strains $\{\varepsilon^e_0\}$ are obtained from a finite element thermal analysis. A detailed derivation of the

element equations for the eight noded isoparametric elastostatic brick element is presented in Appendix E.

7.1.2 Benchmarking of Results

To validate the results obtained by FEECAP an example problem was selected and the thermal and stress analysis results were benchmarked with ANSYS (version 5.0). The example chosen was a single chip surface mounted on a substrate and subjected to thermal static loads and constraints. Figure 13 illustrates the location and magnitude of the applied thermal loads and boundary conditions. The element types chosen in ANSYS for thermal analysis and static analysis were 8-noded brick elements of type SOLID70 and SOLID45 respectively.

Tables 1,2 and 3 show the nodal values obtained from ANSYS and the percent error (%Err) of the results computed by FEECAP with respect to the ANSYS solution. The nodal temperature values were found to be accurate upto six significant digits and the %Err values in the flux components were less than 0.04%. The %Err values in the displacement components were within 0.004% and for the cauchy stress components the maximum %Err value was within 0.005%. These low error values validate the finite element computations performed by FEECAP when compared with ANSYS.



Uniform thickness = 1.0
Thermal loads: applied flux, q1 = 10 on top surface of element 10 (chip)
applied flux, q2 = -8 on top surface of element 1
heat generation Q = 5 by element 7
point heat sources q_p = 6 at nodes 3 and 19
prescribed T = 30 on right face of substrate
all other surfaces are insulated
Static constraints: x, y and z constraints on bottom surface of substrate

Figure 13: Example of a chip surface mounted on a substrate for benchmarking the combined thermal-static stress computations.

Node	T	%Err	q-x	%Err	q-y	%Err	q-z	%Err
1	-955.8	0	-15.71	0.019	-23.34	0.021	5.607	0.002
2	-170.1	0	-15.57	0	-24.28	0.004	3.95	0.008
3	601.2	0	-2.001	0.005	-12.13	0.008	0.579	0.007
4	30	0	11.42	0.035	0	0	0	0
5	211	0	-16.66	0	-33.22	0.012	3.98	0
6	1,044	0	-9.968	0.002	-24.03	0.008	-2.385	0.017
7	1,208	0	10.14	0.01	-8.968	0	-2.55	0.004
8	30	0	23.56	0.017	0	0	0	0
9	2,367	0	2.679	0.008	-30.78	0.016	-0.266	0.015
10	2,233	0	8.686	0.004	-15.22	0.013	-1.058	0.028
11	1,498	0	22.03	0.014	-0.609	0.003	-1.914	0.011
12	30	0	29.36	0	0	0	0	0
13	3,288	0	14.45	0.028	-18.44	0.022	0.132	0.015
14	2,566	0	20.2	0.015	-6.661	0.008	0.525	0.002
15	1,269	0	25.36	0.012	4.586	0.002	0.95	0.005
16	30	0	24.77	0.016	0	0	0	0
17	-983.8	0	-15.88	0.006	-23.5	0.013	5.607	0.002
18	-189.9	0	-15.82	0.006	-24.92	0.02	3.95	0.008
19	598.3	0	-2.199	0.018	-12.44	0.032	0.579	0.007
20	30	0	11.37	0.035	0	0	0	0
21	191.1	0	-17.3	0.023	-33.52	0.006	3.98	0
22	1,056	0	-8.649	0.004	-22.39	0.009	-4.86	0.004
23	1,221	0	7.793	0.005	-7.995	0.003	-4.153	0.007
24	30	0	23.81	0.004	0	0	0	0
25	2,368	0	2.6	0.015	-30.97	0.01	-0.266	0.015
26	2,238	0	8.718	0.001	-15.03	0.02	-1.484	0.027
27	1,508	0	19.5	0.005	-1.069	0.037	-3.829	0.011
28	30	0	29.55	0.007	0	0	0	0
29	3,288	0	14.49	0.021	-18.4	0.022	0.132	0.015
30	2,563	0	20.24	0.01	-6.502	0.005	0.525	0.002
31	1,264	0	25.33	0.004	4.872	0.008	0.95	0.005
32	30	0	24.68	0.004	0	0	0	0
33	1,173	0	-1.649	0.03	-13.69	0.007	-14.76	0.007
34	1,305	0	-1.649	0.03	-3.698	0.008	-10.56	0.038
35	2,263	0	8.344	0.005	-13.69	0.007	-3.191	0.003
36	1,599	0	8.344	0.005	-3.698	0.008	-11.49	0.035

KEY: T = temperature, q_x, q_y and q_z = thermal flux components

$$\% \text{ Err} = \left| \frac{\text{(ANSYS-value) - (FEECAP-value)}}{\text{(ANSYS-value)}} \right| * 100$$

Table 1. Thermal Solution of the Combined Thermal Static Stress Analysis for the Benchmark Example

Node #	U-x (x10)	%Err	U-y (x10)	%Err	U-z	%Err
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	13.5	0.002	8.9	0	-71.8	0
18	-12.2	0.002	-5.4	0	-13.6	0.003
19	-4.9	0	-17.9	0.002	44.9	0
20	12.7	0.002	-3.7	0	1.4	0.001
21	-12	0.002	-16.2	0.003	15	0.003
22	-14.8	0.003	-23.6	0.002	79.8	0.001
23	15.7	0.002	-11.7	0	92.2	0.001
24	13.1	0.001	5.5	0	2.2	0
25	-51.9	0	-43	0.001	177.9	0.001
26	4.5	0.001	-11.1	0.004	170.8	0.002
27	21.8	0.001	1.7	0.001	114.4	0.003
28	14.9	0.003	-5.1	0.001	2.2	0.001
29	-77.6	0.001	76.9	0.001	240.2	0
30	37.4	0	56.2	0	192.7	0.002
31	4.2	0	31.3	0.001	94.9	0
32	19.6	0.002	3.5	0.001	1.2	0.002
33	-36.4	0	-65.4	0.001	118	0.002
34	36.3	0	-41.9	0	135.4	0.002
35	-25.4	0	10.4	0.002	249.8	0
36	63.5	0	24.3	0	168.7	0.002

KEY: U_x, U_y and U_z = displacement components

$$\% \text{ Err} = \left| \frac{\text{(ANSYS-value) - (FEECAP-value)}}{\text{(ANSYS-value)}} \right| * 100$$

Table 2. Displacement Solution of the Combined Thermal Static Stress Analysis for the Benchmark Example

Node #	σ_x	%Err	σ_y	%Err	σ_z	%Err	τ_{xy}	%Err	τ_{yz}	%Err	τ_{xz}	%Err
1	16,960	0.003	16,960	0.003	379	0.001	0	0	1,032	0.003	1,558	0.001
2	2,881	0.001	2,881	0.001	-252	0.001	0	0	-618	0.001	-1,409	0.002
3	-10,720	0.001	-10,720	0.001	-359	0	0	0	-2,062	0.001	-562	0
4	-675	0	-675	0	-346	0	0	0	-431	0.001	1,466	0.002
5	-3,883	0.001	-3,883	0.001	-412	0.001	0	0	-1,872	0.001	-1,382	0.002
6	-18,300	0.003	-18,300	0.003	105	0.004	0	0	-2,725	0	-1,705	0.001
7	-21,190	0.002	-21,190	0.002	86	0	0	0	-1,352	0.003	1,811	0.002
8	-541	0.001	-541	0.001	-31	0.001	0	0	636	0.001	1,513	0.002
9	-41,990	0	-41,990	0	-937	0	0	0	-4,956	0	-5,993	0.001
10	-39,090	0.001	-39,090	0.001	339	0.001	0	0	-1,280	0.001	521	0
11	-26,260	0.001	-26,260	0.001	143	0.001	0	0	193	0.001	2,515	0.002
12	-547	0.001	-547	0.001	-46	0.001	0	0	-587	0	1,718	0.001
13	-59,540	0	-59,540	0	-4,103	0.001	0	0	8,875	0	-8,950	0
14	-45,550	0.001	-45,550	0.001	-1,097	0.001	0	0	6,478	0	4,316	0
15	-22,600	0	-22,600	0	-707	0	0	0	3,613	0	482	0
16	-710	0	-710	0	-426	0.001	0	0	403	0.001	2,257	0
17	16,350	0.003	16,360	0.002	360	0.001	-459	0.001	2,034	0.002	2,230	0.001
18	2,801	0.001	2,592	0.001	-120	0.003	-184	0.002	459	0.001	-736	0
19	-10,020	0.003	-10,170	0.003	52	0.001	247	0.001	-1,517	0.003	-476	0.001
20	194	0.003	2	0.024	118	0.004	168	0.002	-422	0.001	964	0
21	-3,834	0.001	-4,368	0	-298	0.001	-463	0.001	-431	0.001	-635	0
22	-15,800	0.001	-16,180	0.001	-7	0.002	157	0.003	-1,946	0.002	-1,392	0
23	-18,190	0.001	-18,330	0.002	83	0	292	0.001	-1,288	0.004	1,503	0.002
24	-657	0.001	-613	0	-88	0.001	211	0.001	640	0	475	0.001
25	-38,940	0.001	-39,170	0.001	805	0.001	-11	0.001	-3,656	0.001	-6,074	0
26	-32,660	0.001	-32,620	0	1,110	0	507	0.001	75	0	-547	0.001
27	-22,750	0	-22,460	0.002	257	0.001	10	0.002	635	0.001	1,913	0.003
28	-843	0	-707	0	-183	0.003	-41	0.001	-592	0	423	0.001
29	-52,800	0.001	-52,690	0.001	-19	0.001	-535	0	9,594	0	-9,499	0.001
30	-42,660	0.001	-42,050	0.001	855	0	116	0.001	6,730	0.001	3,477	0.001
31	-22,300	0.001	-21,410	0.002	-202	0.001	-507	0.001	3,387	0.001	-622	0
32	60	0	-97	0	-11	0.002	-267	0.002	392	0	1,177	0.004
33	-5,928	0	-5,868	0.001	-93	0	325	0	-2,699	0	-1,870	0.002
34	-6,885	0.001	-7,006	0	-105	0.001	478	0.001	-2,533	0.001	2,107	0
35	-12,920	0.002	-13,160	0.003	290	0.001	234	0	3,273	0.001	-3,584	0.001
36	-8,462	0.001	-8,889	0	86	0.001	387	0.001	2,447	0	3,162	0.001

KEY: σ_x, σ_y, σ_z, σ_{xy}, σ_{yz} and σ_{xz} = Cauchy stress components

$$\% \text{ Err} = \left| \frac{\text{(ANSYS-value) - (FEECAP-value)}}{\text{(ANSYS-value)}} \right| * 100$$

Table 3. Static Stress Solution of the Combined Thermal Static Stress Analysis for the Benchmark Example

7.2 The Four Noded Linear Tetrahedral Element:

Interconnects form a class of objects that have a complex shape and contour. The development of a good mesh of this component poses a problem. A poor mesh will affect the results obtained from the finite element analysis. The four noded tetrahedral element was found to be one of the element types suitable for mesh generation. This element is most commonly used in the automatic mesh generators of commercial software. Additionally this element type was one of the few types supported by the finite element preprocessor in Pro/ENGINEER. The new version of FEECAP (ver. 2.5) has been suitably modified to provide the capability to carry out finite element analysis using meshes made up of four noded linear tetrahedral elements. Fifteen new subroutines have been added and changes have been made to the existing FEECAP routines to accomplish this task.

7.2.1 Element Formulation:

The four noded linear tetrahedral element formulation, unlike the eight noded linear brick element formulation, does not require use of Gaussian quadrature to integrate element matrices and load vectors. Closed form expressions for these quantities are obtained without employing numerical integration or mapping of elements from the real to the natural coordinate space. All calculations are done at nodal points of the element, which reduces the task of extrapolating results from the gauss points as in case of an isoparametric element. The formulation involves the consideration of different cases of boundary conditions for thermal, static and combined thermal and static analysis. Figure 14 shows the nodal numbering scheme followed for the tetrahedral element.

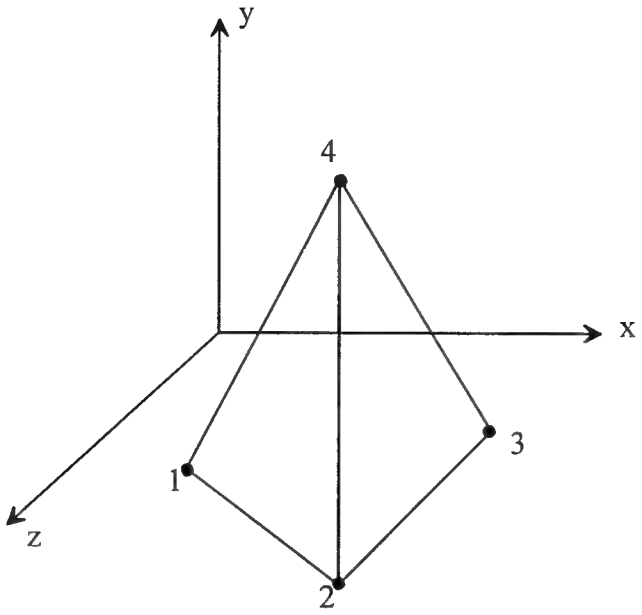


Figure 14: The Tetrahedral Element

The following is a brief summary of the formulation of element equations and the handling of various boundary conditions in FEECAP. Additional mathematical details are given in Appendix I.

7.2.1.1 Thermal Analysis

The mathematical formulation of the finite element method involves the transformation of governing partial differential equations and boundary conditions of a physical system into a set of element algebraic equations. The element equations are assembled into the system equations and then solved simultaneously.

For a three dimensional thermal analysis the following element equations may be written

$$[K^e]\{T^e\} = \{r^e_{\bar{q}_B}\} + \{r^e_Q\} + \{r^e_{\bar{h}}\} \tag{7}$$

where

$$[K^e] = \int_{\Omega^e} [\nabla N^e]^T [k^e][\nabla N^e]\, d\Omega \quad (8)$$

$$\{r^e_{\bar{q}_B}\} = \int_{\Gamma^e} [N^e]^T \bar{q}^e_B\, d\Gamma \quad (9)$$

$$\{r^e_Q\} = \int_{\Omega^e} [N^e]^T Q^e\, d\Omega \quad (10)$$

$$\{r^e_{\bar{h}}\} = \int_{\Gamma^e} [N^e]^T_i \bar{h}^e\, T^e_a\, d\Gamma \quad (11)$$

In the above equations $[K^e]$ is the element conductance matrix, $[k^e]$ is the element material conductivity matrix, $[N^e]$ is the matrix of shape functions, $\{T^e\}$ is the vector of element nodal temperatures, and the $\{r^e\}$ vectors are the element nodal heat loads due to different loading conditions. The first load term, $\{r^e_{\bar{q}_B}\}$, is the due to prescribed flux $\bar{q}_B$ on element face(s), $\{r^e_Q\}$ is due to internal volumetric heat generation Q^e within an element, and $\{r^e_{\bar{h}}\}$ is due to a convection boundary condition with convection coefficient $\bar{h}^e$ and ambient temperature T_a. All the above equations are written at the elemental level (hence the superscript e).

7.2.1.2 Static Analysis

The governing equations for the static analysis with the tetrahedral element are the same integral equations developed for the brick element listed in Equations (2) - (66) on 37page 37. FEECAP also has the capability to handle prescribed nodal displacements for the tetrahedral element. The load vector is adjusted due to these prescribed displacements, if any, before computing the other nodal displacements in the model.

The formulation uses natural coordinates (area and volume coordinates). All calculations and evaluations of integrals are done using these coordinates. The area of an element face in terms of x and y coordinates of its nodes is given by

$$A^e = \frac{1}{2}\det\begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \tag{12}$$

and the volume of an element in terms of the coordinates of its nodes is given by

$$\Omega^e = \frac{1}{6}\det\begin{bmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{bmatrix} \tag{13}$$

Integration of a polynomial in area coordinates over the triangle area is accomplished by the following formula

$$\int_A \xi_1^k \xi_2^l \xi_3^m dA = 2A\frac{k!l!m!}{(2+k+l+m)!} \tag{14}$$

where A is the entire area of the triangular face, k, l, m are non-negative integers and the constraint given by the following equation is satisfied.

$$\xi_1 + \xi_2 + \xi_3 = 1 \tag{15}$$

Similarly for the volume coordinates

$$\int_\Omega \xi_1^k \xi_2^l \xi_3^m \xi_4^n d\Omega = 6\Omega\frac{k!l!m!n!}{(3+k+l+m+n)!} \tag{16}$$

A more detailed treatment of related equations and formulae may be found in Appendix I. A generic formulation for a three-dimensional element is given in Appendix E.

7.2.2 Benchmarking of Results:

All results for the tetrahedral elements have been successfully benchmarked with ANSYS 5.0. Three examples have been presented in this report, one each for the thermal, static and combined analysis. The model used for all the examples is a cube of size 2x2x2 inches (see Figure 15) with thermal conductivity k = 10 Btu/hr °F , coefficient of thermal expansion α = 6.4 (10^{-5}) units?, modulus of elasticity E = 54 (10^6) psi, and Poisson's ratio ν = 0.22. Different boundary conditions

were imposed on the model for each analysis case. The results were compared with those obtained from ANSYS. The corresponding element type used in ANSYS was the collapsed brick element (SOLID70 for thermal analysis and SOLID45 for static analysis).

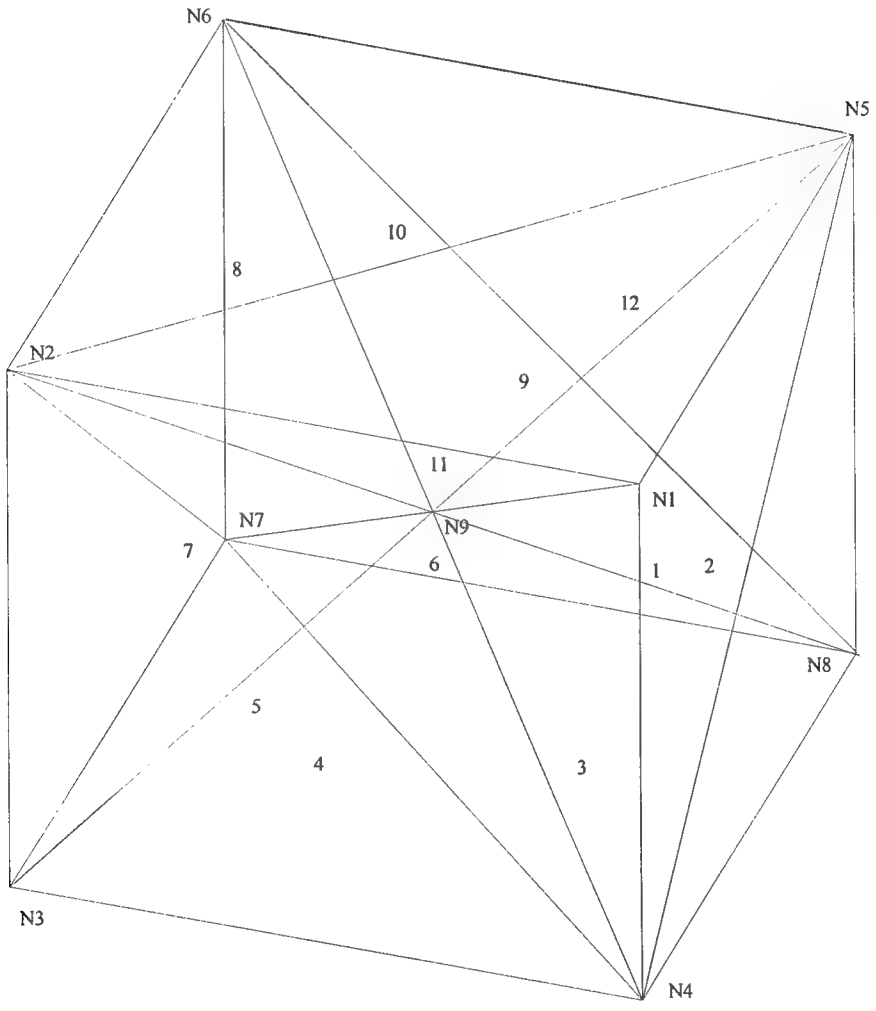


Figure 15: Thermal, static, and combined thermal-static benchmark example for tetrahedral element

7.2.2.1 Thermal Analysis Results for Tetrahedral Element

The following are the comparison results for thermal analysis. The input model consists of prescribed temperatures at nodes 4, 6, 8 (refer to Figure 15 for details) and flux boundary conditions are imposed on two element faces. Table 4 shows the temperature solution for the above problem. As can be seen, the FEECAP nodal temperatures have five significant digit accuracy compared to ANSYS results. Table 5 shows the benchmarked flux results.

NODE	TEMPERATURE	
	ANSYS	FEECAP
1	89.463	89.463
2	92.301	92.301
3	80.891	80.891
4	40	40
5	96.087	96.087
6	80	80
7	90.373	90.373
8	90	90
9	78.319	78.319

Table 4. Temperature Solution Benchmark Comparison for Thermal Analysis (tetrahedral element)

NODE	FLUX-X		FLUX-Y		FLUX-Z		TOTAL FLUX	
	ANSYS	FEECAP	ANSYS	FEECAP	ANSYS	FEECAP	ANSYS	FEECAP
1	43.71	43.7105	-217.79	-217.794	62.642	62.6422	230.8	230.8
2	68.795	68.7948	-104.51	-104.509	33.481	33.4808	129.52	129.522
3	179.7	179.699	-81.806	-81.8057	72.166	72.166	210.22	210.218
4	105.08	105.078	-140.79	-140.792	137.32	137.316	222.98	222.979
5	11.764	11.7636	-125.14	-125.139	64.311	64.311	141.19	141.188
6	-7.2061	-7.20607	-21.365	-21.3654	2.6549	2.65487	22.704	22.7036
7	93.71	93.7105	-43.192	-43.1922	70.025	70.0253	124.7	124.703
8	6.5107	6.51075	-35.082	-35.0822	158.41	158.407	162.38	162.376
9	62.168	62.1681	-95.501	-95.5015	76.254	76.2537	137.11	137.113

Table 5. Flux Solution Benchmark Comparison for Thermal Analysis (tetrahedral element)

7.2.2.2 Static Analysis Results:

The boundary conditions applied to the model for the static analysis case consist of the basic rigid boundary constraints. All three degrees of freedom for node1 are constrained. Node 2 is constrained in Y and Z while nodes 3 and 4 are constrained in Z only. Additionally static load of 100 is applied to nodes 5, 6, 7, 8 in the negative Z direction. Table 6 and Table 7 show the comparison of the displacement and stress solutions respectively for the static analysis.

NODE	U_x		U_y		U_z	
	ANSYS	FEECAP	ANSYS	FEECAP	ANSYS	FEECAP
1	0	0	0	0	0	0
2	8.667E-07	8.667E-07	0	0	0	0
3	7.408E-07	7.408E-07	8.667E-07	8.667E-07	0	0
4	-1.26E-07	-1.26E-07	8.667E-07	8.667E-07	0	0
5	-8.31E-09	-8.31E-09	-8.31E-09	-8.31E-09	-4.33E-06	-4.33E-06
6	6.945E-07	6.945E-07	1.722E-07	1.722E-07	-3.39E-06	-3.39E-06
7	7.491E-07	7.491E-07	8.75E-07	8.75E-07	-4.33E-06	-4.33E-06
8	4.633E-08	4.633E-08	6.945E-07	6.945E-07	-3.39E-06	-3.39E-06
9	3.704E-07	3.704E-07	4.333E-07	4.333E-07	-1.92E-06	-1.92E-06

Table 6. Displacement Solution Benchmark Comparison for Static Analysis (tetrahedral element)

NODE	σ_x		σ_y		σ_z		τ_{xy}		τ_{yz}		τ_{xz}	
	ANSYS	FEECAP	ANSYS	FEECAP	ANSYS	FEECAP	ANSYS	FEECAP	ANSYS	FEECAP	ANSYS	FEECAP
1	1.8949	1.8949	1.8949	1.8949	113.42	113.42	1.3316	1.3316	-1.7387	-1.7387	-1.7387	-1.7387
2	0.24033	0.24033	0.24033	0.24033	104.2	104.2	0.66581	0.66581	-0.69525	-0.69525	0.69525	0.69525
3	1.8949	1.8949	1.8949	1.8949	113.42	113.42	1.3316	1.3316	1.7387	1.7387	1.7387	1.7387
4	0.24033	0.24033	0.24033	0.24033	104.2	104.2	0.66581	0.66581	0.69525	0.69525	-0.69525	-0.69525
5	0.16484	0.16484	0.16484	0.16484	99.311	99.311	-0.27858	-0.27858	-6.2989	-6.2989	-6.2989	-6.2989
6	-1.9877	-1.9877	-1.9877	-1.9877	84.487	84.487	-1.6492	-1.6492	0.3071	0.3071	-0.3071	-0.3071
7	0.16484	0.16484	0.16484	0.16484	99.311	99.311	-0.27858	-0.27858	6.2989	6.2989	6.2989	6.2989
8	-1.9877	-1.9877	-1.9877	-1.9877	84.487	84.487	-1.6492	-1.6492	-0.3071	-0.3071	0.3071	0.3071
9	0	1.14E-14	0	-2.3E-15	100	100	0	5.69E-15	0	-3.33E-15	0	8.448E-15

Table 7. Stress Solution Benchmark Comparison for Static Analysis (tetrahedral element)

7.2.2.3 Combined Thermal Stress Analysis Results:

The combined analysis involves imposing thermal and static boundary conditions on the model. To test the robustness of the formulation, the model was subjected to flux boundary

condition on two faces, convection on two other faces, internal heat generation in three elements and prescribed temperatures on three nodes and even thermal load data as thermal boundary conditions. As static boundary conditions, some degrees of freedom were constrained to zero displacement, two nodes had one degree of freedom each with prescribed displacements greater than zero. Tables 8 through 11 show the comparison of the thermal and static stress solutions obtained by FEECAP with respect to ANSYS. As can be seen from the temperature, flux, displacement and stress solutions, there is a good match of results between ANSYS and FEECAP 2.5.

NODE	TEMPERATURE	
	ANSYS	FEECAP
1	78.514	78.596
2	89.465	89.332
3	76.153	75.313
4	40	40
5	86.076	86.455
6	80	80
7	91.295	91.605
8	90	90
9	77.415	77.524

Table 8. Temperature Solution Benchmark Comparison for Combined Thermal Stress Analysis (tetrahedral element)

NODE	FLUX-X		FLUX-Y		FLUX-Z		TOTAL FLUX	
	ANSYS	FEECAP	ANSYS	FEECAP	ANSYS	FEECAP	ANSYS	FEECAP
1	84.427	83.445	-162.9	-163.216	67.486	69.057	195.49	195.886
2	86.53	85.091	-79.385	-80.422	45.424	47.4331	125.91	126.326
3	163.72	160.857	-83.599	-85.8041	92.753	97.1698	205.91	206.59
4	118.38	116.796	-111.24	-112.126	144.53	146.32	217.44	218.227
5	56.307	55.0152	-72.086	-72.9816	70.83	71.3372	115.69	115.94
6	18.84	18.2317	7.2556	7.52225	14.261	14.0482	24.717	24.2143
7	90.601	90.3017	-37.792	-37.695	85.99	88.205	130.5	131.74
8	31.586	31.6876	-5.4912	-5.9336	162.92	162.379	166.05	165.548
9	80.839	79.7359	-67.377	-68.0383	86.48	87.9061	136.21	136.801

Table 9: Flux Solution Benchmark Comparison for Combined Thermal Stress Analysis (tetrahedral element)

NODE	U_x		U_y		U_z	
	ANSYS	FEECAP	ANSYS	FEECAP	ANSYS	FEECAP
1	0	0	0	0	0	0
2	-1.08E-03	-1.08E-03	0.00E+00	0.00E+00	0	0
3	-1.21E-03	-1.21E-03	-7.96E-04	-7.94E-04	0	0
4	-2.02E-04	-2.03E-04	-1.05E-03	-1.05E-03	0	0
5	3.88E-04	3.88E-04	6.66E-04	6.66E-04	1E-05	1E-05
6	-5.88E-04	-5.89E-04	8.93E-04	8.93E-04	-8.69E-04	-8.69E-04
7	-9.38E-04	-9.40E-04	2.00E-05	2.00E-05	-1.28E-03	-1.28E-03
8	2.45E-04	2.43E-04	-3.72E-04	-3.73E-04	-9.00E-04	-9.01E-04
9	-4.51E-04	-4.53E-04	-1.97E-04	-1.97E-04	-3.58E-04	-3.58E-04

Table 10. Displacement Solution Benchmark Comparison for Combined Thermal Stress Analysis (tetrahedral element)

NODE	σ_x		σ_y		σ_z		σ_{xy}		σ_{yz}		σ_{xz}	
	ANSYS	FEECAP	ANSYS	FEECAP	ANSYS	FEECAP	ANSYS	FEECAP	ANSYS	FEECAP	ANSYS	FEECAP
1	-3870.2	-3887.7	-4562.1	-4578.1	-22722	-22764	2298.8	2316.4	-4652.7	-4657	-1590.2	-1590.9
2	-7814.1	-7725.7	-10589	-10507	-14999	-14924	-132.63	-132.53	-5010.3	-5016	-758.73	-751.93
3	1217.1	1692.2	-5523.4	-5053.7	2181	2653	-1541.9	-1565.3	-6366.6	-6356.6	-1368.2	-1353.6
4	22596	22606	17939	17951	15349	15349	511.07	508.05	-3900.7	-3897.6	-900.39	-901.91
5	-7000.2	-7181.5	-6857.5	-7040	-17669	-17866	1052.4	1066.2	-1202.7	-1204.5	2067.8	2066.5
6	100.88	149.16	-2164.1	-2116	-2166.5	-2120.5	-567.51	-564.83	-3207.1	-3220.2	1634.5	1632.9
7	-5768.2	-5952.4	-13730	-13918	-6805.5	-6993	-1231.3	-1241.1	-4814.1	-4816.3	-382.15	-383.19
8	-5122	-5066.8	-9805.3	-9753.8	-8023.3	-7969.4	-344.17	-343.29	-1086.5	-1086.7	965.89	952.07
9	265.98	220.42	-3450.2	-3497.7	-5830.2	-5884.8	0	-0.020847	-3716.2	-3718.2	0	1.01E-04

Table 11. Stress Solution Benchmark Comparison for Combined Thermal Stress Analysis (tetrahedral element)

In summary, all results for temperatures, fluxes, displacements and stresses agree with those obtained from FEECAP. The results from the thermal and static cases agree *exactly* with the ANSYS results. The results for the combined thermal and static analysis also closely match the results from ANSYS with a maximum error of 1% in the temperature and displacement solution, a maximum of 4.8% error in the flux solution and a maximum of 4% error in the stress solution.

8 Conclusions

To compare the relative merits of various alternative MCM package designs, it is necessary to accurately predict the thermal behavior of small integrated circuit die features, such as field effect transitions, and to accurately predict the mechanical behavior of small, first-level chip-to-substrate interconnects. Through a combination of finite element analyses, submodeling techniques, and the application of specific artificial intelligence techniques, methodologies were developed and implemented in a proof-of-concept software tool, called the Intelligent Multichip Module Analyzer, to predicting Methodologies were developed in this The objective of this

project was to develop rapid, accurate methodologies for prediction of the thermal behavior of integrated circuit features and mechanical behavior of interconnects and to implement these methodologies into the Intelligent Multichip Module Analyzer (***IMCMA***). The combination of a high-level, object-oriented data representation, the blackboard-based problem solving paradigm, and these submodeling methodologies in ***IMCMA*** creates a powerful prototype computer-based tool for comparison of MCM designs.

IMCMA uses the thermal submodeling process to rapidly provide an MCM designer with an estimate of the maximum package temperature of a device. By "zooming" onto critical regions from a macroscopic analysis of an MCM package, ***IMCMA*** avoids the time-consuming process of analyzing large finite element models. With ***IMCMA***, the designer can investigate the thermal characteristics of several candidate MCM designs in the time it would normally take to analyze a single package.

The interconnect submodeling process incorporated in ***IMCMA*** also uses an initial minimum degree of freedom analysis of an MCM model to identify a critical region. Through integration with Pro/ENGINEER and the addition of the tetrahedral element to the finite element solver FEECAP, ***IMCMA*** has the capability to model and analyze complex interconnect geometry. After the creation of a parametric interconnect model, the designer can complete several ***IMCMA*** runs to quickly identify an MCM configuration that produces acceptable stress levels in its interconnects. The complex geometry of MCM interconnects will necessitate large finite element models, but through automatic determination of the critical interconnect and its boundary conditions, ***IMCMA*** will still provide substantial time savings in the design process.

Thermal and interconnect submodeling have both been successfully implemented into ***IMCMA*** to decrease the amount of analysis time needed to complete the MCM design process. The fact that these submodeling methodologies are *automated* provides time savings in addition to those previously discussed. The ***IMCMA*** user only needs to provide input files that contain high level geometric and boundary condition information to run ***IMCMA***. The MCM design engineer doesn't have to perform the tedious, time-consuming tasks involved in the creation of finite element models. The engineer's time can be used more productively in the consideration of other design concerns.

9 Bibliography

(1) *Proceedings of the 1992 IEEE Multi-chip Module Conference*, March 18-20, 1992, Santa Cruz, CA, p. *v*.

(2) R. R. Tummala, E. J. Rymaszewski, editors, *Microelectronics Packaging Handbook*, Van Nostrand Reinhold, New York, NY, 1989, pp. 391-409.

(3) K. Azar, "Thermal Design Considerations for Multichip Module Applications," *Multichip Module Technologies and Alternatives: The Basics*, D. A. Doane, P. D. Franzon, editors, Van Nostrand Reinhold, New York, NY, 1993, pp. 569-613.

(4) R. R. Brown, J. C. Hodge, "Engineering Data Management Issues and Alternative Strategies," *Managing Engineering Data: The Competitive Edge*, edited by R. E. Fulton, 1987 ASME International Computers in Engineering Conference and Exhibition, August 9-13, 1987, NY, NY, pp. 23-26.

(5) K. H. Law, "Conceptual Database Design for Engineering Modeling," *Managing Engineering Data: Emerging Issues*, 1988 ASME International Computers in Engineering Conference and Exhibition, July 31 - August 4, 1987, San Francisco, CA, pp. 19-26.

(6) R. G. G. Cattell, *Object Data Management: object-oriented and extended database systems*, Addison-Wesley, 1992.

(7) D. Corkill, "Blackboard Systems," *AI Expert*, Sept. 1991, pp. 41-47.

(8) D. Corkill, "An Architecture for Intelligent Multichip Module Reliability Analysis," a proposal for Rome Laboratory, NY.

(9) D.P. Finn, J. B. Grimson, N. M. Harty, "An Intelligent Mathematical Modelling Assistant for Analysis of Physical Systems," *Proceedings of the 1992 ASME International Computers in Engineering Conference and Exhibition*, August 2-6, San Francisco, CA, Volume 2, pp. 69-74.

(10) Pepper, D.W and Heinrich, J.C, *The Finite Element Method: Basic Concepts and Applications*, Hemisphere Publishing Corporation, 1992, pp. 153-177.

(11) Zienkiewicz,O.C and Taylor, R.L, *The Finite Element Method*, Fourth Edition, vol. 1, pp. 89- 95.

(12) Cook,R.D, Malkus,D.S and Plesha, M.E, *Concepts and Applications Of Finite Element Analysis*, Third Edition, pp. 147-155.

(13) Shephard, M.S., ``Finite Element Grid Optimization- A Review", *Finite Element Grid Optimization*, M.S.Shephard and R.Gallagher. eds., ASME Special Publication PVP-38, N.Y., 1979.

(14) Babuska, I., O.C. Zienkiewicz, J. Gago, and E.R.A. Oliveira, *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, John Wiley & Sons, New York, 1986.

(15) Ahmed K. Noor, and Ivo Babuska, "Quality Assessment and Control of Finite Element Solutions," *International Journal of Applied Finite Elements and Computer Aided Engineering*, Vol. 3, No. 1, April 1987.

(16) Babuska, I., and W.C. Rheinboldt, ``A *Posteriori* Error Estimations for The Finite Element Method," *International Journal for Numerical Methods in Engineering*, Vol. 12, pp.1597--1615, 1978.

(17) Rheinboldt, W.C., ``Error Estimates for Nonlinear Finite Element Computations", *Computers & Structures*, Vol. 20, No. 1-3, pp. 91-98, 1985.

(18) Babuska, I. and A. Miller, ``The Post-processing Approach in the Finite Element Method - Part 3: *A Posteriori* Error Estimates and Adaptive Mesh Selection", *International Journal for Numerical Methods in Engineering*, Vol. 20, pp.2311--2324, 1984.

(19) Kelly, D.W., J.P. Gago, O.C. Zienkiewicz, and I. Babuska, ``A Posteriori Error Analysis and Adaptive Processes in the Finite Element Method: Part I, Error Analysis", *International Journal for Numerical Methods in Engineering*, Vol. 19, pp. 1593--1619, 1983.

(20) Gago, J.P., D.W. Kelly, O.C. Zienkiewicz, and I. Babuska, ``A Posteriori Error Analysis and Adaptive Processes in the Finite Element Method: Part II, Adaptive Mesh Refinement", *International Journal for Numerical Methods in Engineering*, Vol. 19, pp. 1621--1656, 1983.

(21) Ainsworth, M., J.Z. Xhu, A.W. Craig and O.C. Zienkiewicz, ``Analysis of the Zienkiewicz-Zhu a posteriori error estimator in the finite element method", *International Journal for Numerical Methods in Engineering*, 28, pp. 2161-2174, 1989.

(22) Zienkiewicz, O.C. and Z. Zhu, ``A Simple Error Estimator and Adaptive Procedure for Practical Engineering Analysis", *International Journal for Numerical Methods in Engineering*, Vol. 24, pp.337--357, 1987.

(23) Huang, H.C. and R.W. Lewis, ``Adaptive Analysis for Heat Flow Problems using Error Estimation Techniques", *Sixth International Conference for Numerical Methods in Thermal Problems*, U.K., July 1989, Pineridge Press, Swansea, U.K., 1989.

(24) Lewis, R.W., H.C. Huang, A.S. Usmani and J.T. Cross, ``Finite Element Analysis of Heat Transfer and Flow Problems using Adaptive Remeshing including Application to

Solidification Problems", *International journal for Numerical Methods in Engineering*, Vol. 32, pp.767-781, 1991.

(25) Grosse, I.R., P. Katragadda and J. Benoit, ``An Adaptive Accuracy Based *A Posteriori* Error Estimator", *Finite Elements in Analysis and Design*, pp 70-95, September, 1992.

(26) Bathe and Wilson, ``Numerical Methods in Finite Element Analysis," Prentice-Hall, New Jersey, 1976, Chap. 6

(27) Zienkiewicz, O.C., The Finite Method, Mcgraw-Hill, New York, 3rd edition, 1977

(28) Zienkiewicz, O.C. and Z. Zhu, ``Adaptivity and Mesh Generation", *International Journal for Numerical Methods in engineering,* Vol. 32, pp. 783-810, 1991.

(29) Blacker, T.D. and M.B. Stephenson, ``Paving: A New Approach to Automated Quadrilateral Mesh Generation," *Sandia Report*, SAND90-0249. UC-705, 1990.

(30) Blacker, T.D., J. Jung and W.R. Witkowski, ``An Adaptive Finite Element Technique Using Element Equilibrium and Paving," ASME Paper N0. 90-WA/CIE-2

(31) Grosse, I.R, Katragadda, P and Jog, A, "Knowledge Sources For An Intelligent MCM Analyzer," Final Technical Report, Rome Laboratory Airforce Materiel Command, Griffiss Air Force Base, New York, June 1993

(32) Doane, D. A. and Franzon, P. D., editors, *Multichip Module Technologies and Alternatives: The Basics*, Van Nostrand Reinhold, New York, NY, 1993

Appendix A: A Posteriori Error Estimation and Adaptive Mesh Refinement for Combined Thermal-Stress Finite Element Analysis

Previous sections have dealt with how the critical region of a model can be identified by performing a macro analysis of the complete model and then doing an automatic modeling and analysis of submodels. It is to be noted that there exists inherent discretization errors in the FEA method and hence it is necessary for the user to be aware of the magnitude of such errors and their effect on the solution. Also, in the domain of MCMs, all the components are thermally loaded and statically constrained and hence there is a need for an error estimator which can compute the magnitude of errors for each element under such type of dual-loading.

A. 1. Literature review

In the past few years a significant amount of research has been done in the area of *a priori* and *a posteriori* error estimations for the finite element solution. Various methods have been proposed and some of them have been tested rigorously. For a more complete review of related work done so far in error estimations and adaptive remeshing the reader is referred to [13]-[15]. A majority of the algorithms developed have been for static stress analysis.

Much of the earlier work in residual based error analysis was predominantly mathematical and was done by Babuska and associates [16]-[18]. Based on the residual of the differential equation, the authors derived error bounds for the energy norm of the error and element error indicators were introduced to determine which elements were to be refined. Based on this work Kelly et al [19] proposed a residual based error estimator which used special hierarchical shape functions. These hierarchical shape functions permitted efficient p-based error estimates. A subsequent paper

by Gago et al [20] was published which offered several strategies for using an error estimate to refine a mesh.

Subsequently, Zienkiewicz and Zhu outlined a simple stress-based error estimator and associated adaptivity algorithm. This method involves obtaining a global least-squares fit of the discontinuous finite element stress field with a C^0 continuous stress field. This C^0 continuous stress field is taken as an approximation to the true stress field and the difference between the two stress fields, as measured by the energy or L_2 norm, represents an estimate of the discretization error.

Huang and Lewis [23] developed an error estimation and adaptive technique for linear steady-state conduction problems. In a subsequent paper Lewis et al [24] extended the algorithm to the solution of non-linear transient heat conduction problems. For a measure of the discretization errors, the authors used heat flux which is a function of temperature gradients. For calculating the error norms, they adopted the idea presented by Zienkiewicz and Zhu [22]. Zienkiewicz and Zhu [22] showed that globally smoothed values of stresses are nothing but higher order approximation to the stresses obtained from a finite element analysis, can be used instead of the exact stresses. The requirement that for a near optimal mesh all the elements of the final mesh must contain an approximately equal error was used.

Grosse et al [25] have proposed a stress based finite element error estimator with a new concept of adaptive accuracy and mesh optimality. They have shown that the meshes obtained from the *h*-based adaptivity algorithm were more accurate in the highly stressed regions of the domain with substantially reduced degrees of freedom compared to the adaptive meshes obtained with an uniform accuracy criteria. This was achieved by formulating the accuracy requirement to be a function of the relative behavior of the stress field distribution across the domain. That is, the net

result of this adaptive accuracy criteria is a nonuniform distribution of discretization error with minimum error in maximum stress regions and visa versa.

Here we are concerned with coupled thermal stress analysis problems. For such dual-loading type problems a thermal analysis is first conducted, followed by a static stress analysis. The two analysis are coupled to each other via the thermal strains from the thermal analysis which play the role of initial strains in the static analysis. Thus, the two physical phenomenon are unidirectionally coupled. That is, although thermal behavior directly affects elastostatic behavior via thermal strains, the elastostatic behavior is considered to have a negligible effect on the thermal behavior. This is the case for most thermal-stress engineering problems, but bi-directional thermal-elastostatic coupling can exist for special problems, especially problems involving thermal contact resistance.

When performing the error analysis, the error algorithm can be applied at two different stages. It can be applied to the thermal analysis results or it can be conducted after the static stress analysis is completed. It is important to note that for a given finite element mesh, the thermal analysis solution and the stress analysis solution will yield, in general, different discretization errors throughout the domain. Furthermore, since the stress solution is coupled to the thermal solution (via thermal strains) the thermal solution discretization errors can adversely affect the elastostatic solution. This effect will be undetected in *a posteriori* error analysis of the elastostatic solution.

The most rigorous solution to this problem is to first carry out an iterative *a posteriori* error analysis and adaptive mesh refinement process for the thermal analysis before proceeding to the stress analysis. Then an iterative *a posteriori* error analysis and adaptive mesh refinement is

conducted for the stress analysis portion of the problem. This strategy, however, is prohibitively computationally expensive. Clearly two separate analysis, error estimation and adaptive mesh refinement iteration loops are required. Moreover, for each elastostatic analysis (with the exception of the first one) elemental thermal strains computed from the thermally converged mesh solution must be mapped onto the current elastostatic mesh.

Simpler, more computationally efficient schemes are clearly needed. In this paper we present error estimators for combined thermal and elastostatic analysis and an adaptive remeshing methodology which effectively refines the mesh based on both error estimators in a cost effective manner.

A.2. Error estimator and adaptive accuracy criteria for static stress analysis:

In previous work [25] we introduced a simple error estimator for static analysis, similar to the Zienkiewicz -Zhu error estimator [21], based on the effective stress (von Mises) function σ_v and the L_2 error norm.

$$\|E({}^e\sigma_v)\|_2 = \|{}^e\hat{\sigma}_v - {}^e\sigma_v^*\|_2 = \left[\int_{e\Omega} ({}^e\hat{\sigma}_v - {}^e\sigma_v^*)^2 \, d\Omega\right] \tag{17}$$

where ${}^e\hat{\sigma}_v$ is the effective stress function directly obtained from the finite element displacement and strain solution and ${}^e\sigma_v^*$ is a C^0 continuous `improved' effective stress function computed via stress recovery techniques presented in reference [25].

The relationship of the effective stress function to both distortional strain energy and to the distortional energy failure theory is well known. Also, it has been shown that the L_2 norm of the element's von Mises stress field is proportional to the square root of the element's distortional

strain energy [25]. Hence, a prudent choice is to base the error estimation algorithm on the element's effective stress function and use it as a measure of the element's discretization error.

The elemental error in the norm sense is forced to be less than some small fraction of a reference L_2 norm value:

$$\left\| {}^{e}\hat{\sigma}_v - {}^{e}\sigma_v^* \right\|_2 \le \eta (\|\sigma_v\|_2)_{Ref} \tag{18}$$

where η is small. If the reference value is a *local* (i.e. element based) value, then excessive refinement due to a constant relative accuracy requirement will be imposed for all elements regardless of the stress state of the element. On the positive side, an element based reference value for the L_2 norm bounds the global L_2 error norm in σ_v to be small compared to the global L_2 norm in σ_v. On the other hand, if a single *global* value is taken as the reference, then excessive mesh refinement is avoided, but the global L_2 error norm in σ_v may not be small (i.e.. bounded) compared to the global L_2 norm of σ_v. This possibility can be minimized by adopting a bifunctional adaptive reference norm as follows:

$$(\|\sigma_v\|_2)_{Ref} = \max\left(\| {}^{e}\sigma_v^* \|_2, GRMS\left(\| \sigma_v^* \|_2 \right) \right) \tag{19}$$

where $GRMS\left(\|\sigma_v^*\|_2 \right)$ is the global root mean square of all the elemental norms in the effective stress:

$$\begin{aligned} GRMS(\|\sigma_v^*\|_2) &= \left[\left(\int_V \|\sigma_v^*\|_2^2 dV \right) / V \right]^{\frac{1}{2}} \\ &= \left[\left(\sum_{e=1}^{m} \int_{eV} \|\sigma_v^*\|_2^2 dV \right) / V \right]^{\frac{1}{2}} = \left[\sum_{e=1}^{m} \left(\|{}^{e}\sigma_v^*\|_2^2 \, {}^{e}V \right) / V \right]^{\frac{1}{2}} \end{aligned} \tag{20}$$

Note that Eq. (19) involves a combination of the elemental and global effective stress norms in such a way as to preserve the relative error norms in regions of interest, namely, in elements of high stress states, while eliminating the problem of excessive mesh refinement in regions highly understressed. Figure 16 illustrates the bi-functional adaptive reference norm.

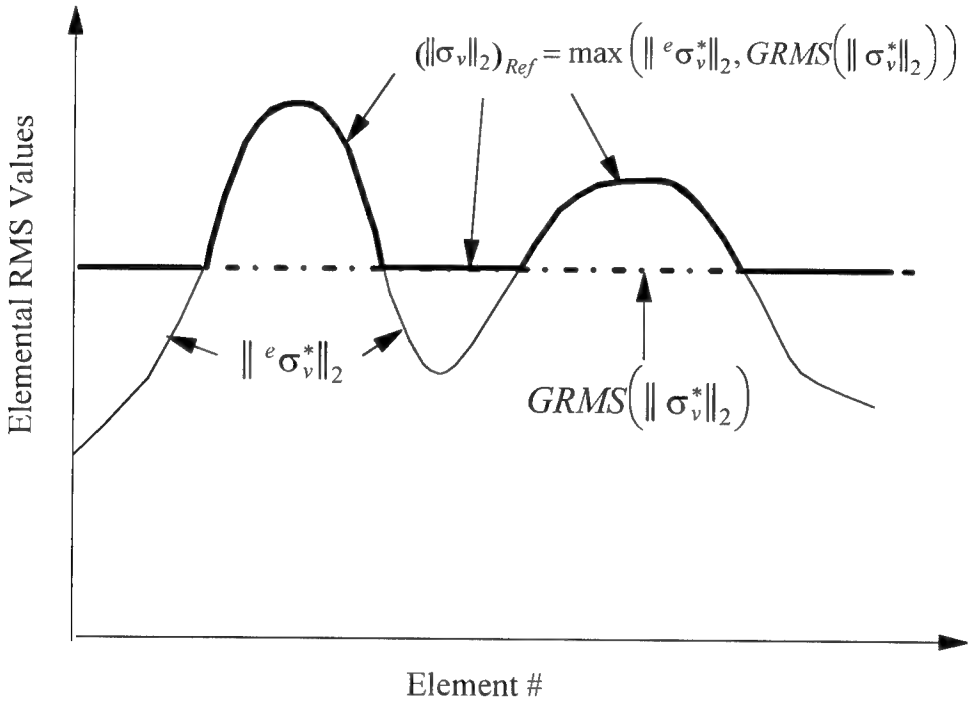


Figure 16: Illustration of the bifunctional reference norm

Substitution of Eq. (19) into (18) results in a bi-functional accuracy requirement based on the L_2 norm of the elements effective stress state compared to the global root mean square value of all the element L_2 effective stress norms:

$$\|^e\hat{\sigma}_v - ^e\sigma_v^*\| \le \eta \ \max\left(\|\,^e\sigma_v^*\|_2, GRMS\left(\|\,\sigma_v^*\|_2\right)\right) \qquad (21)$$

Eq. (21) constitutes a bi-functional adaptive accuracy requirement. In the stress based error estimator and adaptivity algorithms presented by Grosse et al [25] introduced a new concept of adaptive accuracy based on the idea that an *optimal* mesh from an engineering viewpoint is one which adapts according to the element's stress state. This means that the desired accuracy is

achieved at regions of interest with the least number of degrees of freedom. Typically this corresponds to tighter accuracy requirements in regions of higher stress states and relaxed accuracy requirements in regions of lower stress states.

An adaptive accuracy can be extended to the element level by allowing η, the target relative error norm ratio to adapt to the element stress state as follows, by introducing an elemental nominal target accuracy based on the element stress

$$ {}^{e}\eta = \eta\left[\frac{GRMS(\sigma_v^*)}{RMS({}^{e}\sigma_v^*)}\right]^{\alpha} \tag{22} $$

In the above expression η is the nominal target accuracy, $GRMS(\sigma_v^*)$ is the global RMS measure of the effective stress function over the entire domain and $RMS({}^{e}\sigma_v^*)$ is the RMS of the effective stress function distribution for element e:

$$ GRMS(\sigma_v^*) = \left[\int_V (\sigma_v^*)^2 dV/V\right]^{\frac{1}{2}} $$

$$ = \left[\left(\sum_{e=1}^{m} \int_{eV} (\sigma_v^*)^2 dV\right)/V\right]^{\frac{1}{2}} \tag{23} $$

$$ RMS({}^{e}\sigma_v^*) = \left[\left(\int_{eV} ({}^{e}\sigma_v)^2 dV\right)/{}^{e}V\right]^{\frac{1}{2}} \tag{24} $$

Combining the bi-functional adaptive reference norm with the elemental adaptive target accuracy, the accuracy criteria becomes

$$ \|{}^{e}\hat{\sigma}_v - {}^{e}\sigma_v^*\| \le \eta\left[\frac{GRMS(\sigma_v^*)}{RMS({}^{e}\sigma_v^*)}\right]^{\alpha} \max\left(\|{}^{e}\sigma_v^*\|_2, GRMS\left(\|\sigma_v^*\|_2\right)\right) \tag{25} $$

Experimentally, this modification has proven quite promising, limiting the unjustifiably high values for error measurement parameters encountered with the element reference value in understressed regions.

In Eq. (22), α is a constant ≥ 0 which controls the degree of accuracy adaptivity. If $\alpha = 0$, then a nonadaptive accuracy requirement is imposed. It is observed that the effect of Eq. (22) is to increase the accuracy requirement for elements with RMS($^e\sigma_v^*$) greater than the global RMS measure of the smoothed effective stress function. Conversely, the accuracy requirement is decreased for all elements where RMS($^e\sigma_v^*$) < GRMS(σ_v^*).

Practical bounds must be placed on $^e\eta$ computed by Eq. (22) to prevent mesh transitioning problems in an h-refinement meshing scheme. The user specifies *a priori* $^e\eta_{min}$ and $^e\eta_{max}$, the tightest and slackest accuracy requirements, which correspond to the maximum and minimum elemental RMS($^e\sigma_v^*$) values. Hence, from Eq. (22) the accuracy requirements on the element with the highest stressed state and the element with the least stressed state are

$$^e\eta_{min} = \eta \left\lfloor \frac{GRMS(\sigma_v^*)}{\{RMS(^e\sigma_v^*)\}_{max}} \right\rfloor \tag{26}$$

and

$$^e\eta_{max} = \eta \left\lfloor \frac{GRMS(\sigma_v^*)}{\{RMS(^e\sigma_v^*)\}_{min}} \right\rfloor \tag{27}$$

A relationship between $^e\eta_{min}$, $^e\eta_{max}$ and α can be obtained by dividing Eq. (26) by Eq. (27),

$$\frac{^e\eta_{min}}{^e\eta_{max}} = \left[\frac{\{RMS(^e\sigma_v^*)\}_{min}}{\{RMS(^e\sigma_v^*)\}_{max}} \right]^{\alpha} \tag{28}$$

Solving Eq. (28) for a unique value of α

$$\alpha = \left[\frac{Ln(\frac{^e\eta_{min}}{^e\eta_{max}})}{Ln\left(\frac{\{RMS(^e\sigma_v^*)\}_{min}}{\{RMS(^e\sigma_v^*)\}_{max}}\right)} \right] \tag{29}$$

which is based on the user specified values for $^e\eta_{min}$ and $^e\eta_{max}$ and the calculated minimum and maximum elemental **RMS** values in effective stress. Once α is calculated via Eq. (29) it can be

substituted into either Eq. (26) or (27) to determine the value of η. From Eq. (18), a dimensionless parameter called the element error ratio ${}^{e}\xi_S$ for the static stress solution can be defined as:

$$ {}^{e}\xi_S = \frac{\| {}^{e}\hat{\sigma}_v - {}^{e}\sigma_v^* \|_2}{\eta(\|\sigma_v\|_2)_{Ref}} \tag{30} $$

and substituting Eq.(22) in Eq. (30), the final expression for ${}^{e}\xi_S$is given as:

$$ {}^{e}\xi_S = \frac{\| {}^{e}\hat{\sigma}_v - {}^{e}\sigma_v^* \|_2}{\eta\left(\frac{GRMS(\sigma_v^*)}{RMS({}^{e}\sigma_v^*)}\right)^{\alpha}(\|\sigma_v\|_2)_{Ref}} \tag{31} $$

If ${}^{e}\xi_S > 1$, then the element size ${}^{e}h$ is decreased and if ${}^{e}\xi_S < 1$, the element size ${}^{e}h$ is increased. If ${}^{e}\xi_S = 1$ for all elements, then the adaptive accuracy criteria has been met everywhere.

A.3. Error estimator for thermal analysis:

Analogous to the choice of the effective stress function as the basis for measuring the discretization error in static analysis, the thermal heat flux magnitude is chosen for error analysis in the thermal solution. Following the same steps as shown above, the error ratio for thermal analysis is defined as

$$ {}^{e}\xi_T = \frac{\| {}^{e}\hat{q} - {}^{e}q^* \|_2}{\eta\left(\frac{GRMS(q^*)}{RMS({}^{e}q^*)}\right)^{\alpha}\left(\|q\|_2\right)_{Ref}} \tag{32} $$

The heat flux q is given by:

$$ \{q\} = -[K] \underset{\sim}{\nabla} T \tag{33} $$

and the heat flux magnitude is:

$$ q = \left[\{q\}^T \{q\} \right]^{\frac{1}{2}} \tag{34} $$

where [K] is the thermal conductivity matrix of the material and $\underset{\sim}{\nabla} T$ the gradient in the temperature field is given by

$$ \underset{\sim}{\nabla} T = \begin{Bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \\ \frac{\partial T}{\partial z} \end{Bmatrix} \tag{35} $$

Just as in static analysis where mesh refinement is concentrated in regions having high stress gradients, in thermal analysis mesh refinement will be seen to be concentrated in regions of high heat flux gradients.

A.4. Error estimator for coupled thermal and static analysis:

When a body is subjected to both thermal and mechanical loads, the stress-strain relations must include the effects of thermal expansion (and contraction) by subtracting the thermal strains $\{\varepsilon_0\}$ from the total strains $\{\varepsilon\}$. Therefore,

$$\{\hat{\sigma}\} = [E](\{\varepsilon\} - \{\varepsilon_0\}) \tag{36}$$

where

$$\{\varepsilon_0\} = \alpha.\Delta T \lfloor\ 1\ 1\ 1\ 0\ 0\ 0\ \rfloor^T \tag{37}$$

for a three dimensional problem. Here α is the coefficient of thermal expansion for an isotropic material and ΔT is the temperature field relative to a reference temperature at a stress free state. The total strains are due to both the thermally induced and stress induced strains. In the standard finite element formulation the equivalent static loads induced by thermal strains are:

$$\{{}^eR_T\} = \int_{{}^eV} [{}^eB]^T [{}^eE]\{\varepsilon_0\} dV \tag{38}$$

where $[{}^eB]^T$ is the element strain displacement matrix which relates the element strain field to nodal displacements and is given by the gradient of the shape function matrix $[{}^eN]$:

$$\{{}^e\varepsilon\} = [{}^eB]\{{}^ed\} \tag{39}$$
$$\text{where } [{}^eB] = \underset{\sim}{\nabla}\,[N^e]$$

The static analysis system of equations becomes

$$[K_S]\{D\} = \{R_S\} + \{R_T\} \tag{40}$$

Solving Eq. (40) and using Equations (39) and (36) we obtain the stress solution which includes the effects of the thermally induced initial strains.

A.5. Methods of Mesh Refinement:

The most rigorous approach (Method-1) is to carry out an iterative *a posteriori* error analysis and adaptive mesh refinement process for the thermal analysis before proceeding to the stress analysis, and then perform the iterative *a posteriori* error analysis and adaptive mesh refinement process for the static stress analysis portion of the problem. By this method we first refine the mesh to converge to a thermal solution within a user specified accuracy level and then perform further refinement to obtain a stress solution within an acceptable level of accuracy. Since this strategy is computationally expensive, it is not recommended.

A second method (Method-2) is to apply elastostatic error analysis after both the thermal and stress analysis are conducted. This method is not recommended due to the coupling of the stress solution to the thermal solution (via thermal strains) and the effect of the thermal discretization errors in the stress solution which can go undetected in an *a posteriori* error analysis of the elastostatic solution.

Instead, we propose here a simple scheme referred to as the Proposed Method. The thermal analysis is first conducted and an *a posteriori* error analysis on this thermal solution is performed. From this thermal error analysis we obtain the corresponding thermal analysis error ratios ${}^{e}\xi_{T}$. Next, a elastostatic analysis is done using the thermal strains as initial strains and an independent *a posteriori* error analysis is performed on the static solution to obtain the corresponding static analysis error ratios ${}^{e}\xi_{S}$. We thus obtain two sets of elemental error ratios, one from the error analysis conducted on the thermal solution and the other from the error analysis conducted on the

static stress solution (${}^e\xi_S$). These two error ratios for each element signify the accuracy achieved in the thermal solution and the accuracy achieved in the stress solution.

A logical and conservative approach is to use the greater of the two error ratios for a given element as the basis for mesh refinement. Thus, if an element is less accurate in its thermal solution as compared to its stress solution, the element's thermal analysis error ratio is used for recomputing the new element size. Hence the final elemental error ratio (${}^e\xi_C$) due to the combined effect of thermal and mechanical (static) loads is given by the relation:

$$ {}^e\xi_C = \max({}^e\xi_T, {}^e\xi_S) \tag{41} $$

An adaptive mesh refinement is then performed based on these error ratios (${}^e\xi_C$) which take into account the errors in both the thermal and static stress analysis solutions. Intuitively, we expect a faster convergence with this strategy.

A.6. New element size:

From the asymptotic convergence rate criteria, one can derive the following relationship between new and old element sizes [29]:

$$ {}^e_{i+1}h = \frac{{}^e_i h}{{}^e\xi_C \left(\frac{1}{min(p,\lambda)}\right)} \tag{42} $$

where ${}^e_i h$ denotes the element size for the i^{th} mesh, p is the order of the polynomials used in the shape function expansions for the temperature and displacement fields N and λ is the intensity of singularities (if present).

A. 7. Implementation:

The error analysis algorithms were implemented into a FORTRAN-77 code called **FEECAP** which is a three-dimensional **F**inite **E**lement **E**rror **C**ontrol and **A**nalysis **P**ackage for steady

state thermal analysis, static analysis and combined thermal-stress/static analysis. The error analysis algorithm in **FEECAP** estimates the finite element discretization errors and based on these estimates and user specified accuracy requirements, computes the element parameters called error ratios. These error ratios in conjunction with an automatic mesh generator can be used for automatic adaptive mesh refinement. Since the adaptivity algorithms have not yet been developed within the ***IMCMA*** environment, the implementation of the above adaptive remeshing algorithms for the map meshing have been limited to a 2-D automatic mesh generation code called **FASTQ** developed at Sandia National Laboratories. A schematic flowchart of **FEECAP** is shown in Figure 17.

To study the convergence of the global L_2 norm of the errors in the thermal flux field and the effective stress field, we define global error norms as:

$$\phi_q = \left[\sum_{e=1}^{m} \left\| {}^e\hat{q} - {}^e q^* \right\|_2^2 \right]^{\frac{1}{2}} \tag{43}$$

$$\phi_\sigma = \left[\sum_{e=1}^{m} \left\| {}^e\hat{\sigma}_v - {}^e\sigma_v^* \right\|_2^2 \right] \tag{44}$$

The effectivity index (denoted by EI or θ) is a dimensionless parameter which measures the effectiveness of the error estimator and is defined as

$${}^e\theta = \frac{\left\| {}^e\hat{\sigma}_v - {}^e\sigma_v^* \right\|_2}{\left\| {}^e\hat{\sigma}_v - {}^e\sigma_v \right\|_2} \tag{45}$$

where ${}^e\sigma_v$ is the "exact" solution for the element. The effectivity index (EI) can either be used to monitor error measurement for an individual element of the model, or a measure of the effectiveness of the error estimator over the entire domain is given by the global effectivity index:

$$\theta = \left[\frac{\sum_{e=1}^{numc} \left\| {}^e\hat{\sigma}_v - {}^e\sigma_{v2}^* \right\|^2}{\sum_{e=1}^{nume} \left\| {}^e\hat{\sigma}_v - {}^e\sigma_v \right\|_2^2} \right]^{\frac{1}{2}} \tag{46}$$

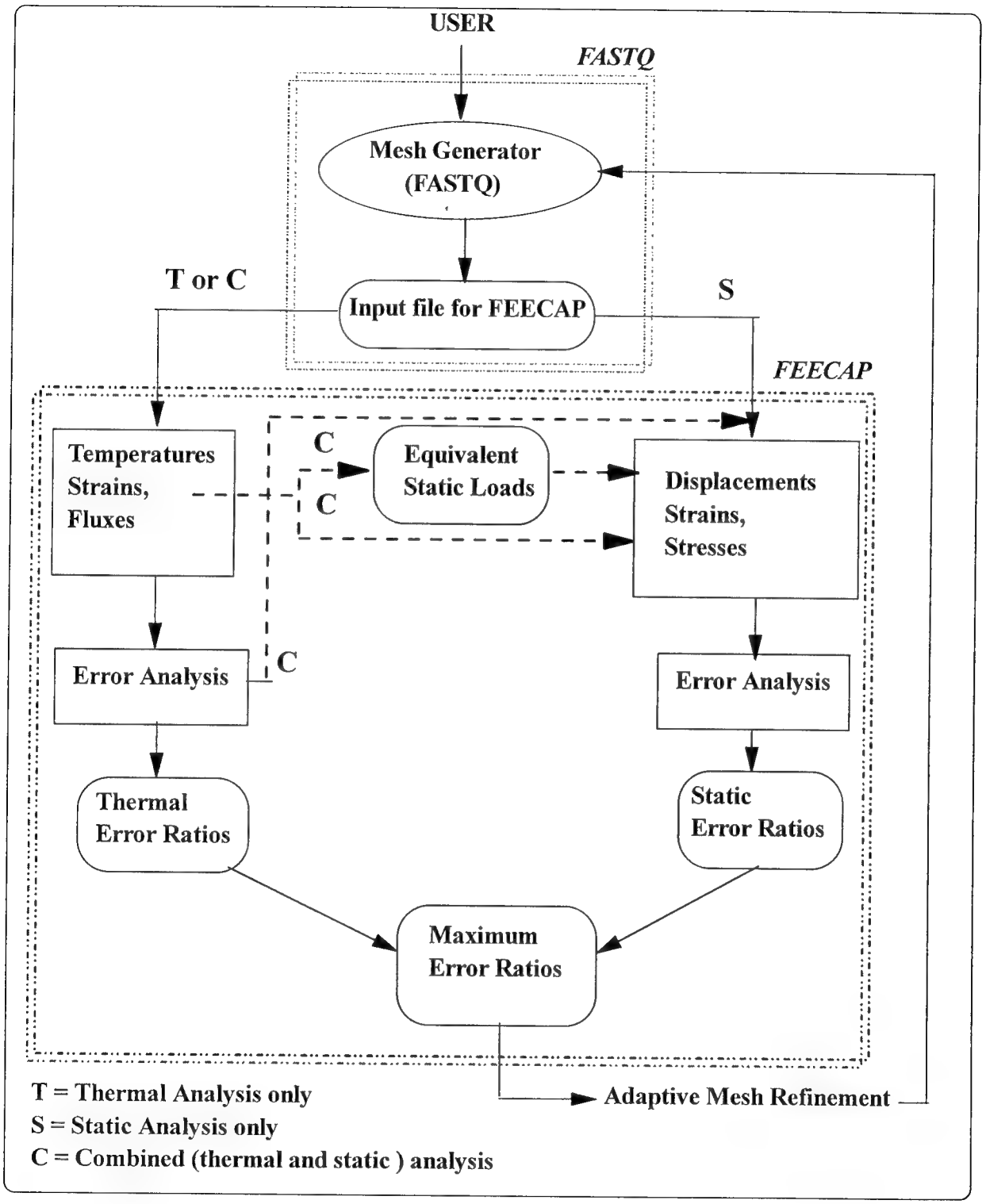


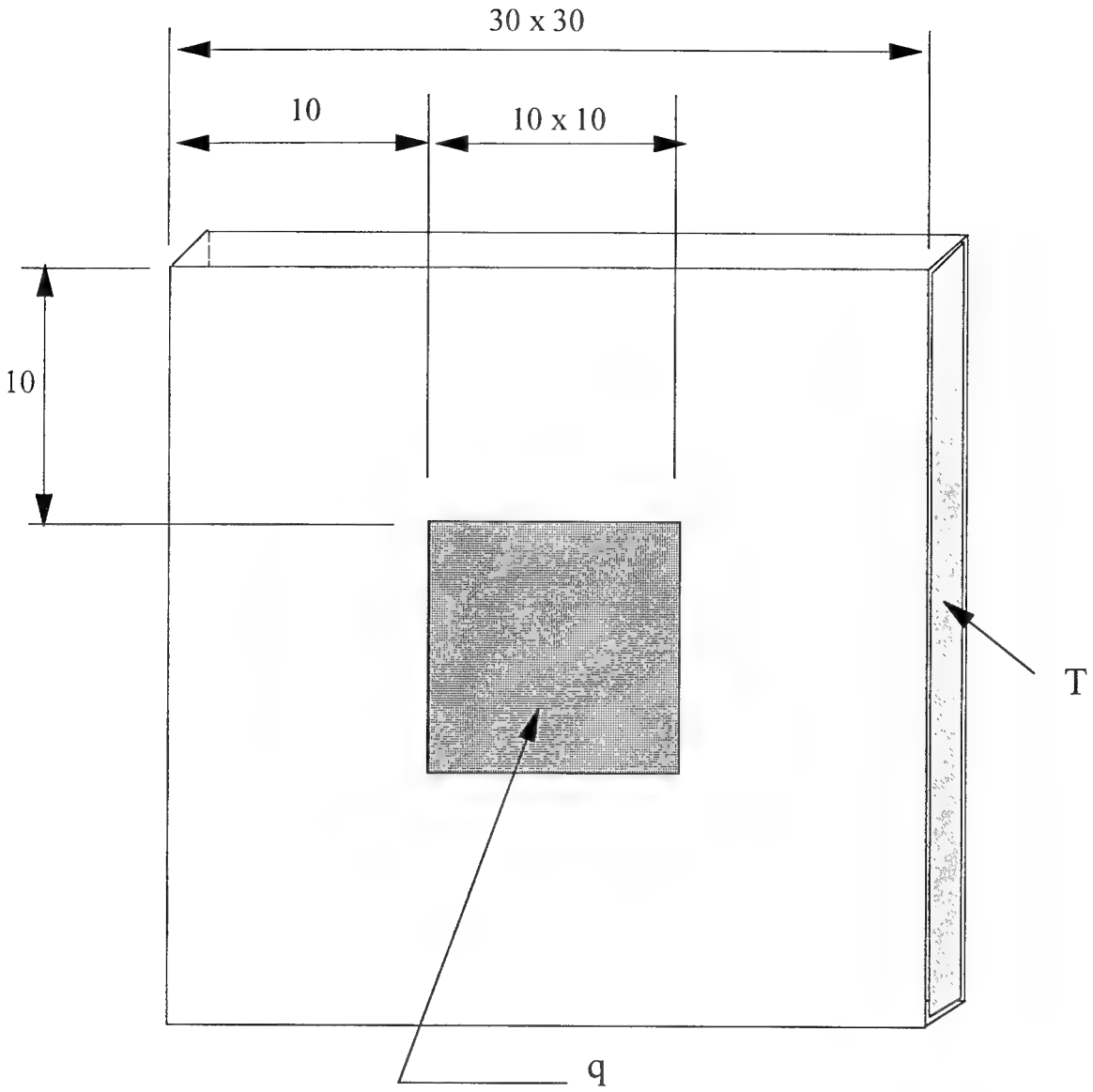
Fig. 17: Flowchart of FEECAP

Eqs. (45) and (46) can be written out in a similar manner for the thermal solution simply by replacing σ_v with the thermal flux vector $\{q\}$ or the scalar flux magnitude function q_m.

A.8. An Example Problem:

A simple example of a chip with prescribed thermal loads and static constraints was selected to demonstrate the convergence and effectiveness of the error algorithm outlined earlier. Figure 18 shows the example chosen with the boundary conditions applied. In ***IMCMA***, nine analysis runs were performed with *default-number-of-elements* set to 1,2,3,....and 9 for each of the runs. The ninth mesh with 729 elements, 1512 active thermal degrees of freedom and 4692 active static degrees of freedom was found to be sufficiently accurate to "truly" represent the thermal flux and stress distributions. Figures (19) and (20) show the convergence of the global error norms (calculated as shown in Eqs. (43) and (44)) versus the active degrees of freedom.

Figs. (21) and (22) show the convergence of the global effectivity index, computed by Eq. (46) for the thermal and static stress solutions. These graphs illustrate the decay of the maximum and minimum effectivity indices across the whole domain of the chip example for each mesh.



Uniform thickness = 1.0
Thermal loads: applied flux, q = 0.03
prescribed T = 30
all other surfaces are insulated
Static constraints: x, y and z constraints on 4 corners of bottom surface

Figure 18: Example of a chip with applied thermal and static boundary conditions used for benchmarking the combined error estimation algorithm.

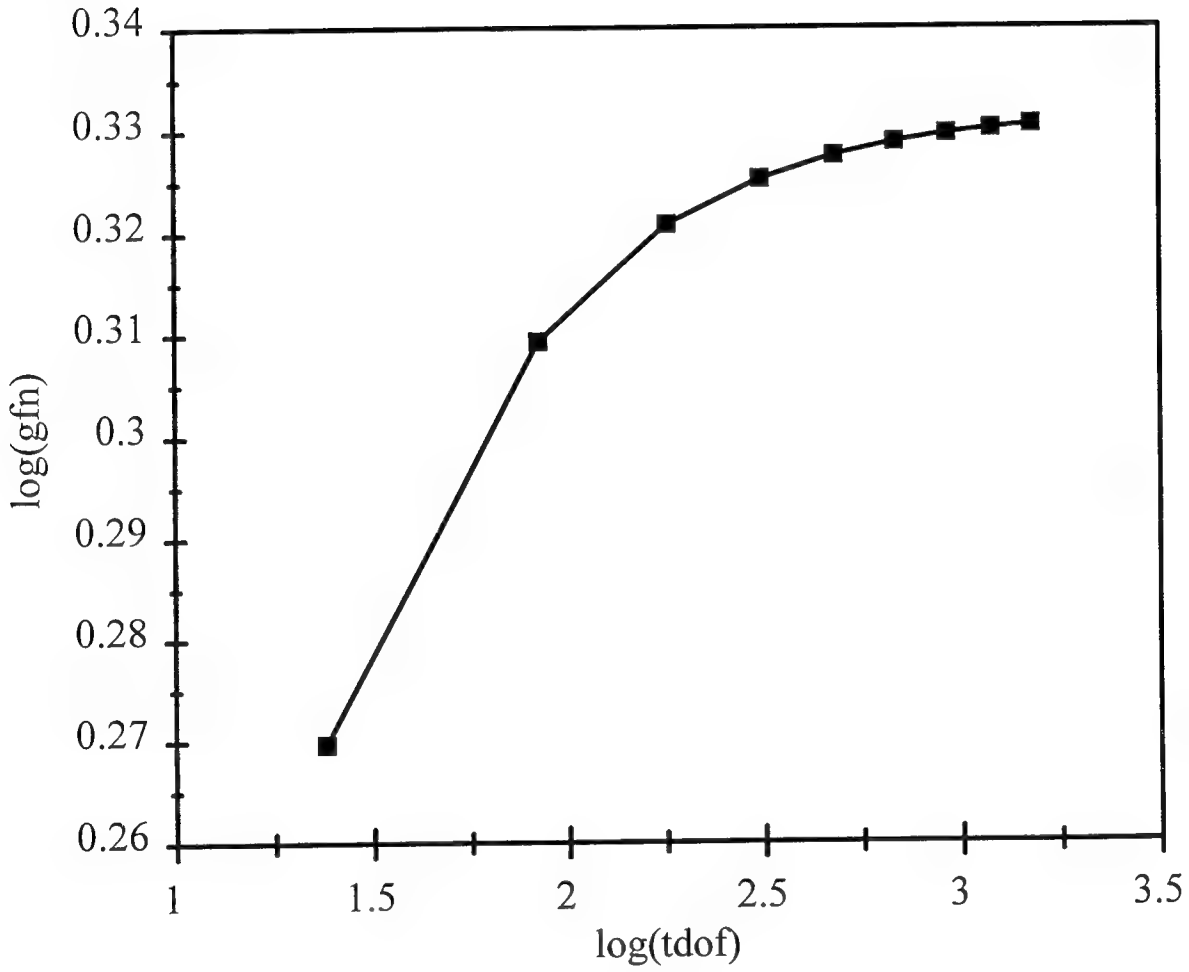


Figure 19. Convergence of global flux norm (gfn) versus thermal degrees of freedom (tdof)

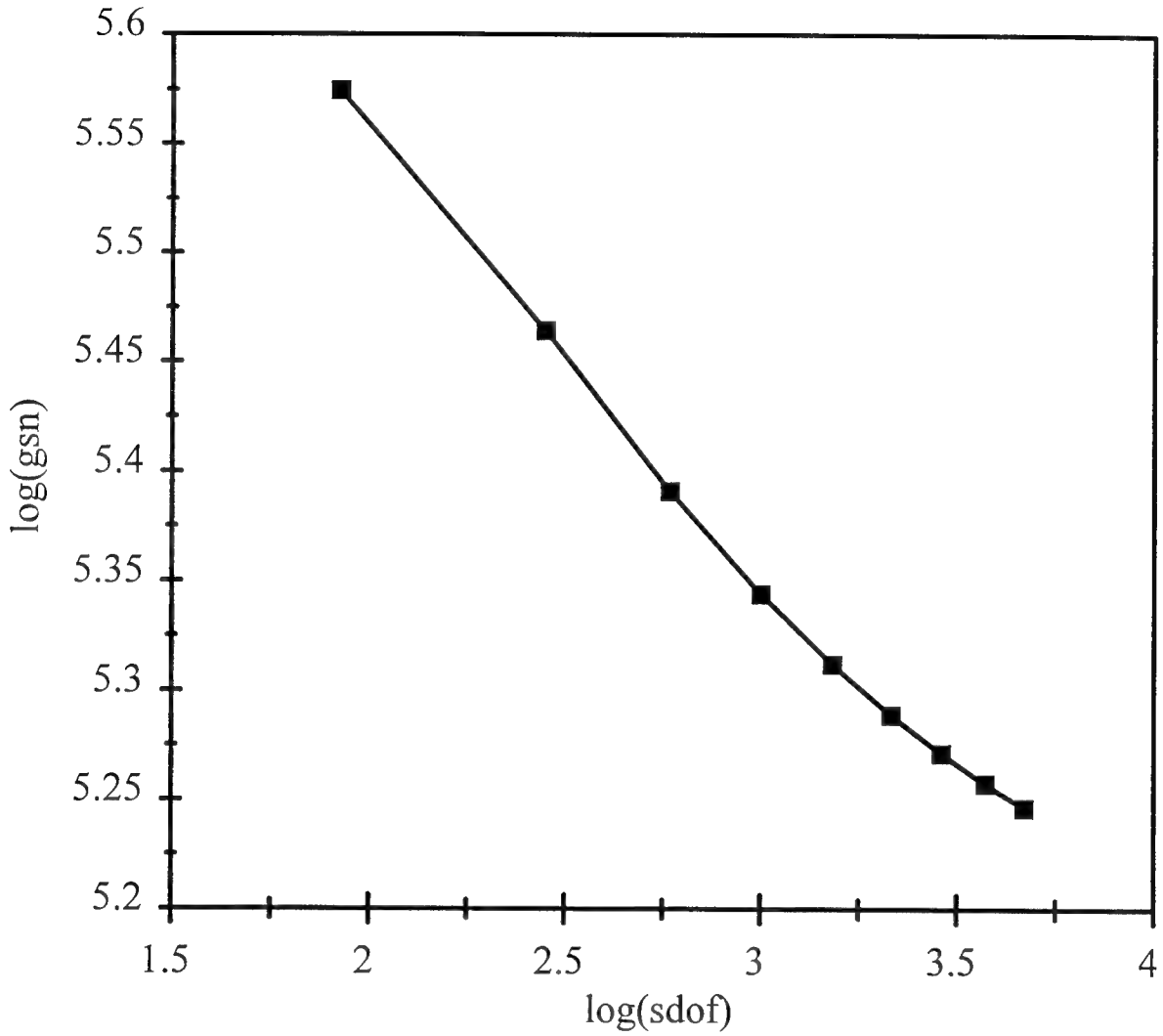


Figure 20. Convergence of global stress norm (gsn) versus static degrees of freedom (sdof)

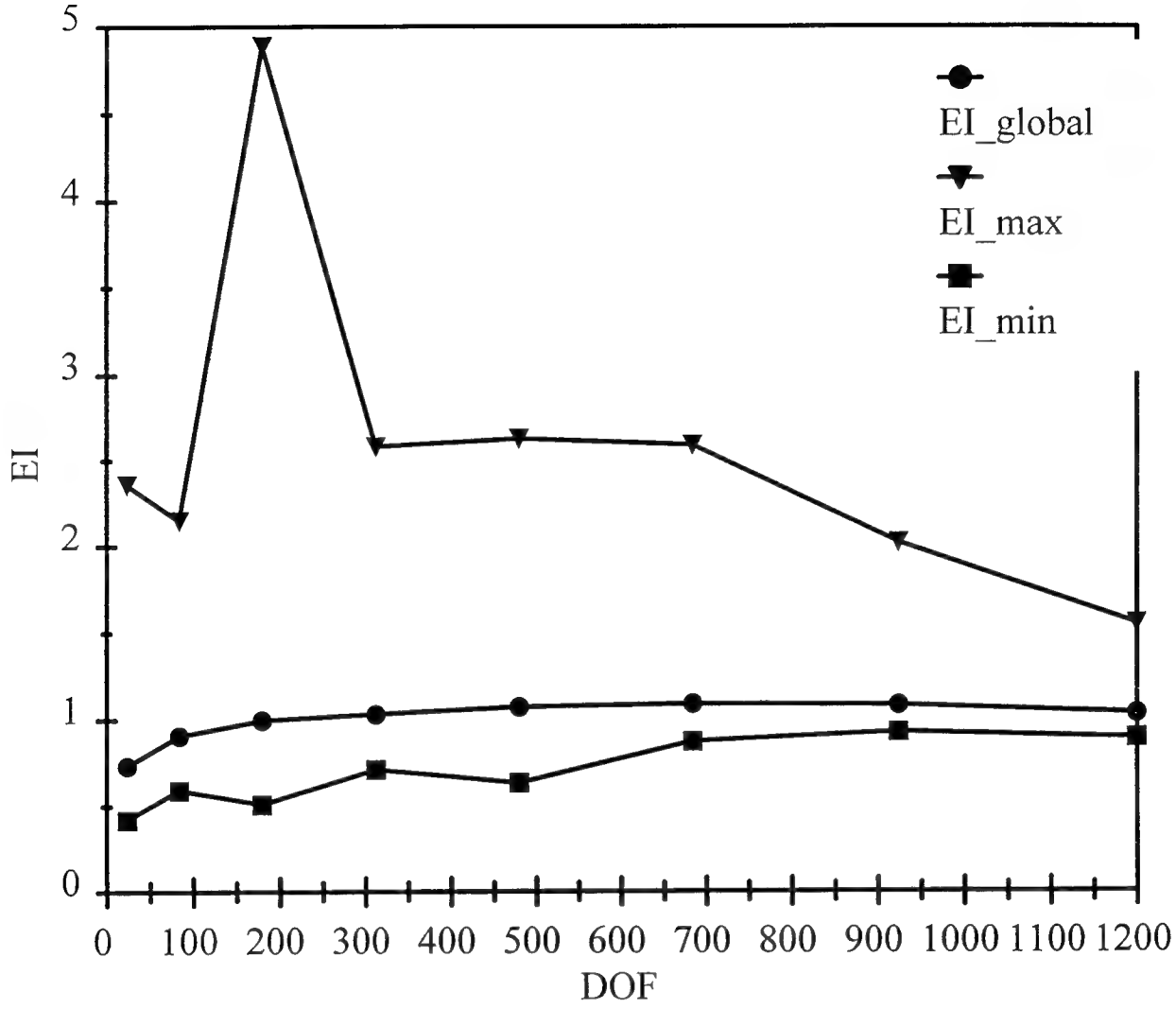


Figure 21: Convergence of effectivity index (EI) versus degrees of freedom (DOF) for thermal solution

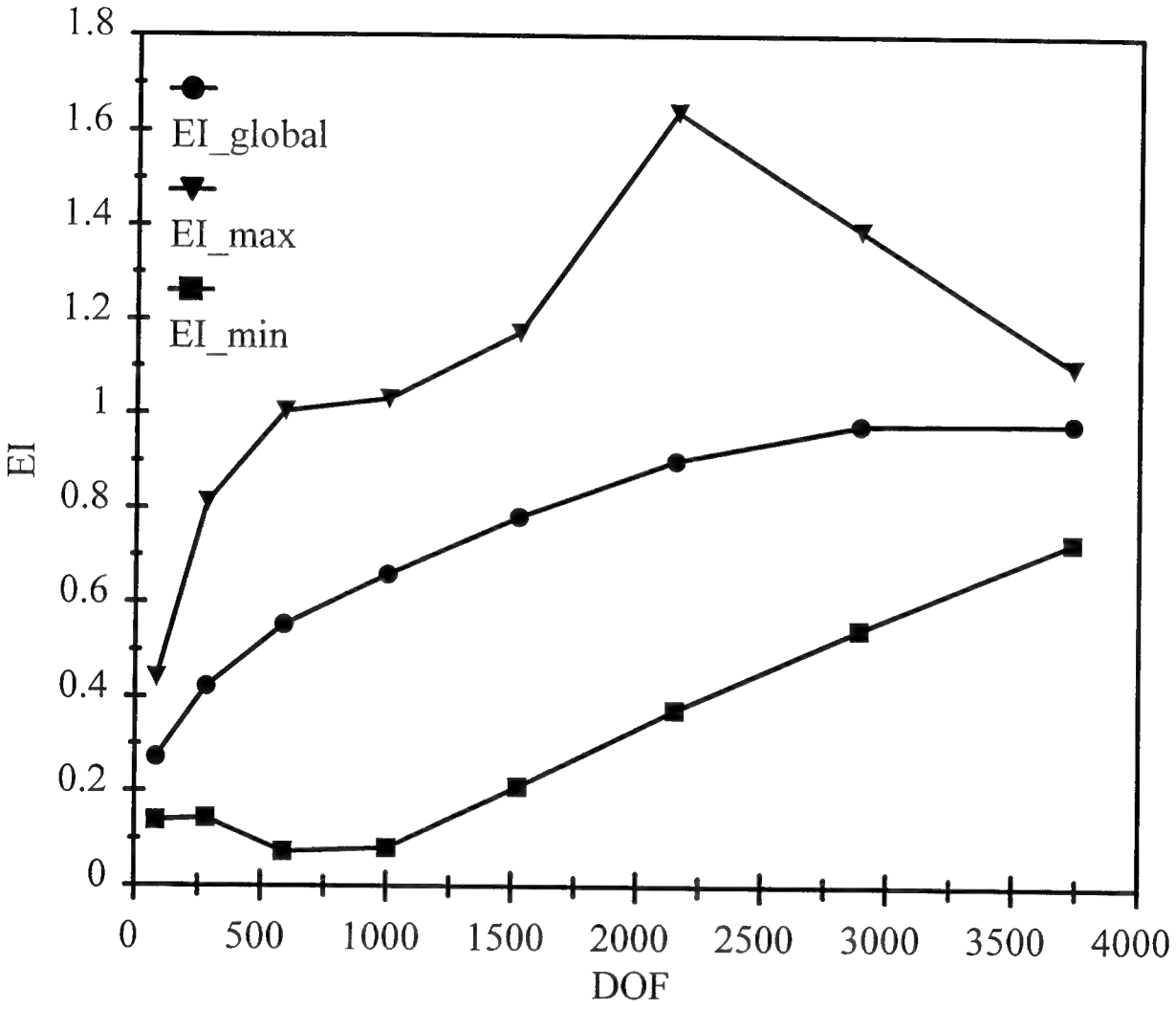


Figure 22: Convergence of effectivity index (EI) versus degrees of freedom (DOF) for the stress solution

Appendix B. Running Thermal Submodeling in *IMCMA*

RUN-*IMCMA*

Five keywords have been implemented in the ***run-imcma*** function to carry out submodeling within the Intelligent Multichip Module Analyzer (***IMCMA***) software system. These keywords are optional arguments to the ***run-imcma*** function.

thermal-submodel This keyword is used to run thermal submodeling in ***IMCMA***. This includes the macroscopic model analysis and two submodel analyses.

Example: (run-***imcma*** "example" :submodel t)

The submodeling process will be completed from input information found in example.lisp. The "t" parameter added to the :submodel option stands for "true".

interconnect This keyword is used to identify the critical interconnect on an MCM, and to analyze the Pro/Engineer mesh of the interconnect. The displacements at the critical interconnect locations are applied as boundary conditions to the mesh of the interconnect.

Example: (run-***imcma*** "example2" :interconnect)

The critical interconnect will be identified from the model and interconnect specifications in example2.lisp. The Pro/Engineer mesh will then be analyzed and the results imported into the ***IMCMA*** database.

die-attach-material and **die-attach-thickness**

These keywords must be used together to define the two die attach parameters. The die attach parameters are used in two different ways. If the *equivalent-k* keyword option is used in ***run-imcma***, the parameters are used to calculate equivalent thermal conductivities. If the *submodel* option is used, the die attach is actually modeled in the

first submodel. The parameters are then used to define a die attach component and to bind a material to this component

Example: (run-***imcma*** "example3" :submodel t
:die-attach-material ':epoxy
:die-attach-thickness 0.05)

An epoxy die attach with a thickness of 0.05 will be included in the first submodel analysis. The material must be defined in the database or an error will result.

equivalent-k This keyword instructs the system to create materials with equivalent thermal conductivities for the macroscopic model analysis. The effects of the chip and the die attach are represented by giving the chip a material with an equivalent conductivity. The *die-attach-material* and *die-attach-thickness* keywords must be included in the ***run-imcma*** command for this option to run.

Example: (run-***imcma*** "example4" :submodel t :equivalent-k t
:die-attach-thickness 0.05
:die-attach-material ':epoxy)

Chip materials will be replaced with materials having a conductivityequivalent to the chip/die-attach combination for the macroscopic analysis. The original materials are returned for the subsequent submodel analyses, and a die attach is modeled for the first submodel analysis.

COMPONENTS AND THERMAL SUBMODELING

When running a thermal submodel analysis, both components and 2nd-subcomponents must be defined by using the *defcomponent* and *def2ndsubcomponent* macros in the input file. (See example input file in Appendix C) The *defcomponent* calls define the basic geometry of the MCM. The small heat sources found on the chip surfaces must be defined as 2nd-subcomponents by using the *def2ndsubcomponent* macros. These 2nd-subcomponents are used to define point heat

sources for the first submodel, and the 2nd-subcomponents in the critical region are used in the second submodel analysis. If no 2nd-subcomponents are defined and the submodel option of run-***imcma*** is specified, an error will be signaled at some point in the analysis.

Appendix C. Example of a Thermal Submodeling Input File

```
;;;;    -*- Mode:COMMON-LISP; Package:IMCMA; Base:10 -*
;;;;    *-* File: /user2/sheehy/imcma2/verif.lisp *-*
;;;;    *-* Edited-By: sheehy *-*
;;;;    *-* Last-Edit: Wed Apr 13 16:07:41 1994 *-*
;;;;    *-* Machine: hinault.ecs.umass.edu *-*

;;;;
****************************************************************
;;;;
****************************************************************
;;;;      *
;;;;      *        THERMAL SUBMODELING VERIFICATION EXAMPLE
;;;;      *
;;;;
****************************************************************
;;;;
****************************************************************
;;;
;;;     Written by:    Michael Sheehy
;;;                    UMass Amherst
;;;
;;;     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
;;;
;;;     3-18-94 File created.
;;;
;;;     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

(in-package "IMCMA")

;; This must come first:

(define-device "SUBMOD MCM"
  :filename "submod-ex"
  :size (15.00 8.00 1.00))

;; Materials come next:

(defmaterial :silicon :tk .0539 :alpha 0.57e-5)
(defmaterial :aluminum :tk .15 :alpha 0.27e-5)

;; The substrate:
```

```
(defcomponent :SUBSTRATE-1 :substrate
  :size (15.00 8.00 1.00)
  :prescribed-temperature-surfaces ((:bottom 20))
  :material :aluminum)

;; The chips:

(defcomponent :CHIP-1 :chip
  :size (4.0000 4.000 .25)
  :x (+ 4.00)
  :y (+ 4.00)
  :z (+ 1.00)
  :xy-alignment :centered
  :power-dissipation 10.341
  :material :silicon)

 (defcomponent :CHIP-2 :chip
  :size (4.0000 4.0000 .25)
  :x (+ 11.00)
  :y (+ 4.00)
  :z (+ 1.00)
  :xy-alignment :centered
  :power-dissipation 5.2
  :material :silicon)

;; The heat sources:

(def2ndsubcomponent :HS-1 :heat-source
  :size (0.5 0.5 0.05)
  :x (+ 2.75)
  :y (+ 2.75)
  :xy-alignment :centered
  :power-dissipation 1.5
  :material :silicon)

 (def2ndsubcomponent :HS-2 :heat-source
  :size (0.5 0.5 0.05)
  :x (+ 4.00)
  :y (+ 2.75)
  :xy-alignment :centered
  :power-dissipation 1.2
  :material :silicon)

 (def2ndsubcomponent :HS-3 :heat-source
  :size (0.5 0.5 0.05)
```

```
  :x (+ 5.25)
  :y (+ 2.75)
  :xy-alignment :centered
  :power-dissipation 0.7
  :material :silicon)

(def2ndsubcomponent :HS-4 :heat-source
  :size (0.5 0.5 0.05)
  :x (+ 2.75)
  :y (+ 4.00)
  :xy-alignment :centered
  :power-dissipation 2.0
  :material :silicon)

(def2ndsubcomponent :HS-5 :heat-source
  :size (0.5 0.5 0.05)
  :x (+ 4.00)
  :y (+ 4.00)
  :xy-alignment :centered
  :power-dissipation 1.7
  :material :silicon)

(def2ndsubcomponent :HS-6 :heat-source
  :size (0.5 0.5 0.05)
  :x (+ 5.25)
  :y (+ 4.00)
  :xy-alignment :centered
  :power-dissipation 0.4
  :material :silicon)

(def2ndsubcomponent :HS-7 :heat-source
  :size (0.5 0.5 0.05)
  :x (+ 2.75)
  :y (+ 5.25)
  :xy-alignment :centered
  :power-dissipation 0.8
  :material :silicon)

(def2ndsubcomponent :HS-8 :heat-source
  :size (0.5 0.5 0.05)
  :x (+ 4.00)
  :y (+ 5.25)
  :xy-alignment :centered
  :power-dissipation 1.3
  :material :silicon)
```

```
(def2ndsubcomponent :HS-9 :heat-source
  :size (0.5 0.5 0.05)
  :x (+ 5.25)
  :y (+ 5.25)
  :xy-alignment :centered
  :power-dissipation 0.741
  :material :silicon)
;;; -----------------------------------------------------------------
;;;  End of File
;;; -----------------------------------------------------------------
```

Appendix D. Implementation of Thermal Submodeling

The following section details the new files and knowledge sources created for thermal submodeling, as well as the changes in the existing knowledge sources. All changes are documented in the code, but these descriptions offer an overview of the changes.

***imcma*-system.lisp**

This file was modified to include two submodeling knowledge sources, *model-submodel-transition-ks* and *submodel-2nd-submodel-transition-ks*, and one additional file, ***imcma-interpolation***, in the ***IMCMA*** package. This ensures that the new files are loaded with the rest of ***IMCMA***.

***imcma*-main.lisp**

The function *run-**imcma***, which starts the operation of ***IMCMA***, is modified to contain additional options which are needed for submodeling. These options are discussed in Appendix B. The *init-**imcma**-internal* function has been modified to initialize some new global variables created for the submodeling process.

***imcma*-model-units.lisp**

The existing unit class definitions in this file have been changed through the addition of new slots, and the class names have been modified to accommodate the storage of model and submodel unit classes on different blackboard spaces. Additional submodel unit classes have also been added to this file.

***imcma*-unit-mappings.lisp**

New unit mappings for submodel unit classes and submodel event definitions are added to this file.

***imcma*-preamble.lisp**

Many global variables used in submodeling are defined in ***imcma-preamble***, and all submodel blackboard and space definitions are included in this file.

exodus-utilities.lisp

The function *read-analysis-results* has been modified to accommodate the submodel node and element unit classes.

***imcma*-graphics.lisp**

Display functions have been modified to display submodel unit instances, and methods have been altered to accommodate model and submodel unit classes, through the specification of their superclasses.

***imcma*-interpolation.lisp**

Imcma-interpolation is a new file, and contains an interpolation subroutine that is called between the levels of submodeling. Also included are functions needed for finding the elements to interpolate from.

KNOWLEDGE SOURCES: The following files contain ***IMCMA***'s knowledge sources. All of the files with the exception of *input-model-ks.lisp* have functions that have been modified to accommodate submodel unit classes. Some functions have new arguments which specify the unit classes used at the current level of modeling. These arguments are necessary to carry out operations on the correct unit classes. In addition, global variables containing unit class specifications are used in retrieval functions in most or all of the knowledge sources. All the files, again with the exception of *input-model-ks*, contain multiple knowledge source definitions for the

multiple levels of modeling. These KS definitions can be consolidated into one for each file if necessary.

input-model-ks.lisp

The major modification to this file is the addition of the *defsubcomponent* and *def2ndsubcomponent* functions, which are currently called from the ***IMCMA*** input file to define unit instances of submodel components. Another small modifications is made to *input-model-ks* to trigger the *model-submodel-transition-ks*, which will be activated after all other modeling knowledge sources are carried out.

complete-model-ks.lisp

A function has been added to *complete-model-ks* called *change-material-numbers*. This function prevents exodus component block errors caused by materials that are not used at the current level of modeling. Another small modification has been made to trigger the correct knowledge sources for the current level of modeling.

generate-mm-regions-ks.lisp

The only addition is a conditional statement that triggers the correct knowledge source for the current level of modeling.

generate-2d-mesh-ks.lisp

No modifications besides the changes described above for all knowledge sources.

extrude-component-ks.lisp

The *find-wells* function has been modified to add extrusion layers for submodels. Without this modification, the submodels are created with only one layer of elements through the thickness. The code can easily be changed to obtain the desired number of layers.

combine-3d-meshes-ks.lisp

In the first level of submodeling, point heat sources are created to approximate the heat added by the small 2nd subcomponent heat producing regions. The function *link-prescribed-surfaces-to- elements* has been modified to identify the heat producing regions in the submodeling region, and to create point heat sources to approximate the effect of these regions.

analyze-3d-mesh-ks.lisp

Some extensive additions have been made to the function *invoke-feecap* in order to carry out the interpolations necessary for prescribing boundary conditions to a submodel. *Invoke-feecap* has also been changed to identify the materials used at the current level of submodel, and to write only the necessary materials to the FEECAP input file.

model-submodel-transition-ks.lisp

This is a completely new knowledge source that performs many operations necessary for the commencement of the first level of submodeling. The display dimensions are changed for graphical display of the submodel and graphics commands are called to produce the display. The region to be submodeled is identified, and the macroscopic model elements which will be used in interpolation are identified and stored in a global variable. The submodel component unit instance is created, and if the adhesive is to be included in the submodel, the die attach unit instance is also created. In the function *init-**imcma**-submodel-internal*, several global variables are set for submodel use.

submodel-2nd-submodel-transition-ks.lisp

This knowledge source performs many of the same functions as *model-submodel-transition-ks*, except for the initiation of the second submodeling process. The major difference is that a die

attach is not included in the second submodel. Also, any 2nd subcomponents that exist on the blackboard and are not used in the second submodel are identified and removed from the database.

Appendix E. A Generic 3-D Element Formulation

The related equations and a brief description of the eight noded isoparametric brick element formulation for elastostatic analysis is discussed in this appendix.

The Galerkin's weighted residual statement for element *e* is written as

$$\oint_{\Gamma^e} N_i^e \{R_\Gamma^e\}\, d\Gamma + \int_{\Omega^e} N_i^e \{R_\Omega^e\} d\Omega = 0, \quad i = 1, 2, ..., N^e \tag{47}$$

where Γ^e is the element surface area, Ω^e is the element volume, N_i^e is the element shape function associated for the ith node of the element, N^e is the number of nodes per element, and $\{R_\Gamma^e\}, \{R_\Omega^e\}$ are the boundary and domain residual vector functions respectively given by

$$\{R_\Omega^e\} = \nabla \bullet [\sigma^e] + \{f^e\} \tag{48}$$

$$\{R_\Gamma^e\} = \{{}^{(n)}\overline{\tau}\} - \{\hat{n}\} \bullet [\sigma^e] \tag{49}$$

In the above equations $\{\hat{n}\}$ is the unit normal vector, $\{{}^{(n)}\overline{\tau}\}$ is the component of the externally applied traction vector in the normal direction on the element surface, $\{f^e\}$ is the body force acting on the element and $[\sigma^e]$ is the element stress tensor.

The domain and boundary residual functions given by Equations (48) and (49) represent a lack of satisfaction of differential equilibrium within the element and on the element surface. If these residual functions are identically equal to zero through out the element volume and everywhere on the element surface, then the differential equilibrium equations are completely satisfied in this part of the system domain and the stress tensor $[\sigma^e]$ is the exact stress tensor admitted by the theory of elasticity.

Substitution in the Galerkin weighted residual statement gives

$$\oint_{\Gamma^e} N_i^e(\{{}^{(n)}\bar{\tau}^e\} - \{\hat{n}\} \bullet [\sigma^e])d\Gamma + \int_{\Omega^e} N_i^e(\nabla \bullet [\sigma^e] + \{f^e\})d\Omega = 0, \quad i = 1, 2, ..., N^e \tag{50}$$

Integrating by parts the first term of the domain integral yields

$$\oint_{\Gamma^e} N_i^e \{{}^{(n)}\bar{\tau}^e\} d\Gamma - \int_{\Omega^e} \nabla N_i^e \bullet [\sigma] d\Omega + \int_{\Omega^e} N_i^e \{f^e\} d\Omega \tag{51}$$

The first term represents the element nodal loads due to an externally applied traction vector $\{{}^{(n)}\bar{\tau}^e\}$ along the element edges. Upon assembly, this vector will contain zero load values for all nodes for which there are no externally applied traction vectors on interelement boundaries *and* no externally applied point nodal loads. The last term represents the element nodal loads to the body force vector $\{f^e\}$. Body forces in elasticity are most commonly due to gravity or inertia forces in which a dynamic analysis has been converted into a static analysis by D'Alembert's principle.

By letting i increment over all nodes of the element in Eqn. (49) and using nontensor notation, we can write the set of element equations in the form

$$\int_{\Omega^e} [B^e]^T \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{Bmatrix} d\Omega = \{r_{\bar{\tau}}^e\} + \{r_f^e\} \tag{52}$$

where

$$[B^e]_i = \begin{bmatrix} N_{i,x}^e & 0 & 0 \\ 0 & N_{i,y}^e & 0 \\ 0 & 0 & N_{i,z}^e \\ N_{i,y}^e & N_{i,x}^e & 0 \\ 0 & N_{i,z}^e & N_{i,y}^e \\ N_{i,}^e z & 0 & N_{i,x}^e \end{bmatrix} \tag{53}$$

$$\{r_{\bar{\tau}}^e\} = \oint_{\Gamma^e} [N^e]^T \{\bar{\tau}\} d\Gamma \tag{54}$$

$$\{r_f^e\} = \int_{\Omega^e} [N^e]^T \{f\} d\Omega \tag{55}$$

$$[N^e] = \begin{bmatrix} N_1^e & 0 & 0 & N_2^e & 0 & 0 & \dots & N_{N^e}^e & 0 & 0 \\ 0 & N_1^e & 0 & 0 & N_2^e & 0 & \dots & 0 & N_{N^e}^e & 0 \\ 0 & 0 & N_1^e & 0 & 0 & N_2^e & \dots & 0 & 0 & N_{N^e}^e \end{bmatrix} \tag{56}$$

The term on the left hand side of the above equation gives rise to the stiffness matrix of the element. We now introduce the constitutive relations

$$\{\sigma\} = [E](\{\varepsilon\} - \{\varepsilon\}_0) \tag{57}$$

The matrix [E] is and the strain displacement relations

$$\{\varepsilon^e\} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \end{Bmatrix} = \begin{Bmatrix} u,_x \\ v,_y \\ w,_z \\ u,_y + v,_x \\ v,_z + w,_y \\ u,_z + w,_x \end{Bmatrix} = \begin{bmatrix} 1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1 \\ 0&1&0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0&1&0 \\ 0&0&1&0&0&0&1&0&0 \end{bmatrix} {}^{xyz}\nabla \begin{Bmatrix} u^e \\ v^e \\ w^e \end{Bmatrix} \tag{58}$$

where $u, v,$ and w are the x, y, and z components of the displacement field, respectively and ${}^{xyz}\nabla$ is the gradient operator with respect to the x,y,z coordinate system.

For assumed displacement field based finite elements, the element displacement field $\{u^e,v^e,w^e\}^T$ is interpolated from element nodal displacements $\{u^e_i,v^e_i,w^e_i\}^T$ using the element shape functions (interpolating polynomials) N_i^e:

$$\begin{Bmatrix} u^e \\ v^e \\ w^e \end{Bmatrix} = \begin{Bmatrix} \sum_{i=1}^{N^e} N_i^e u_i^e \\ \sum_{i=1}^{N^e} N_i^e v_i^e \\ \sum_{i=1}^{N^e} N_i^e w_i^e \end{Bmatrix} = \begin{bmatrix} N^e \end{bmatrix} \begin{Bmatrix} u_1^e \\ v_1^e \\ w_1^e \\ \cdots \\ u_{N^e}^e \\ v_{N^e}^e \\ w_{N^e}^e \end{Bmatrix} = [N^e]\{d^e\} \tag{59}$$

and the matrix $[N^e]$ has been previously defined in Eqn. (56). We use Eq. (58) to substitute for the displacment field into Eq. (57) and obtain

$$\{\varepsilon^e\} = [B^e]\{d^e\} \tag{60}$$

and the matrix [B^e] in Eqn. (60) is identical to the [B^e] matrix defined in Eqs. (53) and (54). Using Eqn. (60Eqn. (57), the following expression is obtained for the element stress vector.

$$\{\sigma^e\} = [E^e](\{B^e\}\{d^e\} - \{\varepsilon_0\}) \tag{61}$$

At this point we can now substitute for the stress vector given by Eqn. (61) into Eqn. (52) to obtain

$$\int_{\Omega^e} [B^e]^T [E^e](\{B^e\}\{d^e\} - \{\varepsilon_0\})\, d\Omega = \{r_{\bar{t}}^e\} + \{r_f^e\} \tag{62}$$

Rearranging this expression produces the final element equations:

$$[K^e]\{d^e\} = \{r_{\bar{t}}^e\} + \{r_f^e\} + \{r_{\varepsilon_0}^e\} \tag{63}$$

where

$$[K^e] = \int_{\Omega^e} [B^e]^T [E^e][B^e] d\Omega \tag{64}$$

$$\{r_{\varepsilon_0}^e\} = \int_{\Omega^e} [B^e]^T [E^e]\{\varepsilon_0\} d\Omega \tag{65}$$

All that remains to do is to adopt the isoparametric formulation, since it makes the integrals easier to evaluate numerically, leads to compatible elements, and makes the element shape functions independent of the element geometry.

Since the shape functions are written in terms of the natural coordinates ξ, η and ρ we must convert the gradient operator with respect to *x*, *y* and z to the gradient operator with respect to the natural coordinates:

$$^{xyz}\nabla = [\Gamma^e]^{\xi\eta\rho}\nabla \tag{66}$$

where

$$[\Gamma^e] = [J^e]^{-1} = \begin{bmatrix} J_{11}^e & J_{12}^e & J_{13}^e \\ J_{21}^e & J_{22}^e & J_{23}^e \\ J_{31}^e & J_{32}^e & J_{33}^e \end{bmatrix}^{-1} \tag{67}$$

and

$$
\begin{aligned}
J^e_{11} &= \frac{\partial x^e}{\partial \xi} = \sum_{i=1}^{N^e} \frac{\partial N^e_i}{\partial \xi} x^e_i \\
J^e_{12} &= \frac{\partial y^e}{\partial \xi} = \sum_{i=1}^{N^e} \frac{\partial N^e_i}{\partial \xi} y^e_i \\
J^e_{13} &= \frac{\partial z^e}{\partial \xi} = \sum_{i=1}^{N^e} \frac{\partial N^e_i}{\partial \xi} z^e_i \\
J^e_{21} &= \frac{\partial x^e}{\partial \eta} = \sum_{i=1}^{N^e} \frac{\partial N^e_i}{\partial \eta} x^e_i \\
J^e_{22} &= \frac{\partial y^e}{\partial \eta} = \sum_{i=1}^{N^e} \frac{\partial N^e_i}{\partial \eta} y^e_i \\
J^e_{23} &= \frac{\partial z^e}{\partial \eta} = \sum_{i=1}^{N^e} \frac{\partial N^e_i}{\partial \eta} z^e_i \\
J^e_{31} &= \frac{\partial x^e}{\partial \rho} = \sum_{i=1}^{N^e} \frac{\partial N^e_i}{\partial \rho} x^e_i \\
J^e_{32} &= \frac{\partial y^e}{\partial \rho} = \sum_{i=1}^{N^e} \frac{\partial N^e_i}{\partial \rho} y^e_i \\
J^e_{33} &= \frac{\partial z^e}{\partial \rho} = \sum_{i=1}^{N^e} \frac{\partial N^e_i}{\partial \rho} z^e_i
\end{aligned}
\tag{68}
$$

$[J^e]$ is the 3x3 Jacobian transformation matrix from the x,y,z system to the natural coordinate system.

Most of the formulation discussed above contains equations that are similar to those of the tetrahedral element formulation (Appendix I). The equations used in this appendix are generic equations and may be used for any 3-D element type.

Appendix F. Determining the Critical Interconnect

***IMCMA*-interconnect.lisp:**

This module was developed to define the unit-class for interconnects. The various slots for the unit-class interconnect were

- type - indicating type of interconnect (e.g.: wirebond, TAB etc.) material
- indicating type of material of interconnect.
- start-point - location of start-point of interconnect on the MCM.
- end-point - location of end-point of interconnect on the MCM.
- start-displacements - stores the displacements (interpolated from the macroscopic mesh) at the start-point of the interconnect
- end-displacements - stores the displacements (interpolated from the macroscopic mesh) at the end-point of interconnect.
- static-temperatures - stores the temperatures (interpolated from the macroscopic mesh) at the start-point of interconnect end-ic-temperatures
- stores the temperatures (interpolated from the macroscopic mesh) at the end-point of interconnect
- relative-displacement - stores the relative displacement of the interconnect.

input-model-ks.lisp:

This module was modified to include the defmacro for the interconnect according to the unit-class definition. The defmacro "definterconnect " enables the user to input the attributes of each interconnect and correspondingly each instance of the interconnect is saved onto the blackboard.

***IMCMA*-preamble.lisp :**

The blackboard path for each instance of the interconnect is defined and space is created to store the instances of the interconnect. The path for the interconnect instance is "model 3d-model interconnects". Any information regarding the interconnects is retrieved by specifying this path.

feecap.f :

Feecap had to be modified to write out the nodal displacements in the exodus format

***IMCMA*-model-units.lisp** :

New slots had to be defined for the nodal-displacements unit class to store the displacements at each node. Moreover, since a combined analysis has to be performed for the identification of critical interconnect a unit class POINT-DISPLACEMENT-CONSTRAINT was added. Thus while defining the components such as chip and substrate in the input file, point displacement constraints could be specified on the macroscopic model as (lower-left-front-corner (1 0 0)), where 1 indicates constraint and 0 indicates free. Hence, enabling displacement constraints to be specified, combined analysis could thus be carried out.

exodus-utilities.lisp :

This module had to be modified to read the displacements of each node written in the exodus format. The displacements of each node are then stored onto the blackboard.

***IMCMA*-unit-mappings.lisp**

In this module an event is defined to enable to trigger the critical interconnect knowledge source.

critical-interconnect-ks.lisp:

This knowledge source was developed to identify the critical interconnect. Once the macroscopic mesh analysis is performed the nodal displacements of each element is available and can

be retrieved from the blackboard. The location of each interconnect is identified with the top elements of the macroscopic model and displacements at the nodes of the element which encompasses the start point or the end point of the interconnect are *interpolated* onto the start or end point of the interconnect respectively. The interpolation routine is discussed below

Interpolation: The MCM is modeled with rectilinear eight-noded brick elements. A two-dimensional interpolation is carried out for the interpolation of displacements from the nodes of the element to the start or end point of the interconnect since the start or the end point of the interconnect lie on a face of the encompassing element and not in the interior of an element. This simplifies our interpolation from three dimensional to two dimensional. The shape functions or interpolating functions are calculated for the four nodes of a face of the element in terms of the isoparametric coordinates. The shape functions are given as:

$$\begin{aligned} N_1(\xi,\eta) &= \tfrac{1}{4}(1-\xi)(1-\eta) \\ N_2(\xi,\eta) &= \tfrac{1}{4}(1+\xi)(1-\eta) \\ N_3(\xi,\eta) &= \tfrac{1}{4}(1+\xi)(1+\eta) \\ N_4(\xi,\eta) &= \tfrac{1}{4}(1-\xi)(1+\eta) \end{aligned} \tag{69}$$

As the real space coordinates of the interpolating point are known, these shape functions are then substituted in terms of the isoparametric coordinates in the following equations:

$$\begin{aligned} x_I &= N_1x_1 + N_2x_2 + N_3x_3 + N_4x_4 \\ y_I &= N_1y_1 + N_2y_2 + N_3y_3 + N_4y_4 \end{aligned} \tag{70}$$

where x_I and y_I are the real space coordinates of the interpolating point *I*, N_1 to N_4 are the shape functions of the element which encloses the interpolating point, and x_1 to x_4, y_1 to y_4 are the x and y coordinates of the nodes of the face of the element respectively. The isoparametric coordinates ξ, η are then solved knowing the coordinates of the start point or end point of the interconnect

from the two functions

$$\begin{aligned} 4x_I - (x_1 + x_2 + x_3 + x_4) &= (-x_1 + x_2 + x_3 - x_4)\xi + (-x_1 - x_2 + x_3 + x_4)\eta + (x_1 - x_2 + x_3 - x_4)\xi\eta \\ 4y_I - (y_1 + y_2 + y_3 + y_4) &= (-y_1 + y_2 + y_3 - y_4)\xi + (-y_1 - y_2 + y_3 + y_4)\eta + (y_1 - y_2 + y_3 - y_4)\xi\eta \end{aligned} \tag{71}$$

The root (ξ_I, η_I) which satisfies Eqn. (71) and the condition

$$\begin{aligned} -1 \le \xi \le 1 \\ -1 \le \eta \le 1 \end{aligned} \tag{72}$$

is substituted back into the shape functions and the displacements at point I (i.e. either the start or end point of the interconnect) is interpolated from nodal displacements using the formula

$$\begin{aligned} u_I &= N_1(\xi_I, \eta_I)\, u_1 + N_2(\xi_I, \eta_I)\, u_2 + N_3(\xi_I, \eta_I)\, u_3 + N_4(\xi_I, \eta_I)\, u_4 \\ v_I &= N_1(\xi_I, \eta_I)\, v_1 + N_2(\xi_I, \eta_I)\, v_2 + N_3(\xi_I, \eta_I)\, v_3 + N_4(\xi_I, \eta_I)\, v_4 \\ w_I &= N_1(\xi_I, \eta_I)\, w_1 + N_2(\xi_I, \eta_I)\, w_2 + N_3(\xi_I, \eta_I)\, w_3 + N_4(\xi_I, \eta_I)\, w_4 \end{aligned} \tag{73}$$

Once the displacements are interpolated to the start and end point of each interconnect the relative displacement is calculated for each interconnect. The relative displacement between the start and end points of the interconnect is determined by:

$$\Delta d = \sqrt{(u_s - u_e)^2 + (v_s - v_e)^2 + (w_s - w_e)^2} \tag{74}$$

The interconnect with the largest relative displacement is identified and stored on the black-board while the remaining interconnect instances are deleted from the blackboard. Further sub-modeling of the critical interconnect can be performed by retrieving the instance from the blackboard.

***IMCMA*-main.lisp** :

This module has been modified to include an option for identifying the critical interconnect. While invoking run-***imcma*** with the option - interconnect true - the CRITICAL-INTERCONNECT-KS.LISP is triggered.

With these new additions and modifications the identification of a critical interconnect is fully functional.

Appendix G. Modeling Tips for the Interconnect

The user is free to use the wide variety of features available in PRO-ENGINEER for modeling the interconnect. A few methods used by the developers are suggested below:

Method 1

- Create default datum planes and coordinate axes.
- Read in the datum points from the *points.dat* file, which is automatically created during the execution of the PRO-MODEL-WRITE-KS.
- Use the *solid pipe feature* (for wirebonds) to create the interconnect joining the points using a spline curve.
- Create the thermocompression (for wirebonds) on one end face of the interconnect as a *protrusion* feature. Rounding all edges is a good idea.
- Copy this feature on the other end face by specifying alternate datum planes and points using the *copy feature* option in PRO-E.
- Enter the FEM module of PRO-E and create the tetrahedral mesh using the *tet-mesh* option. Do *not* use the default global element length for the meshing process.
- Output the mesh to an ANSYS file *bond.ans* choosing the linear structural element type from the menu.
- Exit PRO-E.

Method 2

- Repeat first two steps from method 1.
- Use the protrusion-sweep menu to create the interconnect.

- First sketch the spline curve starting with the start point and ending with the end point to define the profile of the interconnect. Dimension the spline with reference to the default datum planes.
- Sweep the section of the interconnect along the profile.
- Repeat steps 4 through 8 of method 1.

Note: While any method may be used to model the interconnect, the following is mandatory:

- The start and end points always lie at the "z-level" at which the nodes, which are prescribed displacements, lie.
- The start point and end point must be read from the points.dat file. The choice of the intermediate points is left to the user.
- The start point is the location on the interconnect where the section of the interconnect starts. The flattened region/the region where the interconnect is fixed to the chip or substrate must extend away from this section. The same rule applies to the end point. The length of this region is the thermocompression length.

Appendix H. Interconnect Modeling and Analysis - Knowledge Sources Added and Modified

The following knowledge sources have been added to ***IMCMA*** for the interconnect modeling and analysis:

(1) ***pro-model-ks.lisp :*** This knowledge source is responsible for interfacing ***IMCMA*** with PRO-ENGINEER. When executed, it opens up a PRO-ENGINEER window. Some user instructions are also output on the LISP window.

(2) ***tetra-3d-mesh-ks.lisp:*** This knowledge source reads in the mesh data from the ANSYS file and instantiates nodes, elements on the blackboard and also links the interconnect instance with the mesh data.

(3) ***tetra-mesh-analyze-ks.lisp:*** This knowledge source determines the nodes that lie at the start and end of the interconnect and assigns the corresponding displacements to these nodes. Further it accesses the mesh data, material information and boundary conditions from the blackboard and writes out the FEECAP input file.

(4) ***read-results-ks.lisp:*** This knowledge source reads in the post analysis results obtained from FEECAP and the results are written back to the blackboard. The data read includes nodal displacements and stresses.

The following knowledge sources have been modified in the ***IMCMA*** system to integrate the modeling of the interconnect with the other existing modules:

(1) ***imcma-system.lisp:*** An addition has been made to the list of knowledge sources to include all the new knowledge sources written for interconnect modeling.

(2) ***imcma-main.lisp:*** A counter for the number of interconnect nodes and elements has been introduced.

(3) ***imcma-preamble.lisp:*** This knowledge source is modified to include blackboard spaces for the interconnect, interconnect-3d-node and interconnect-3d-element and also to define the global variable for the number of nodes with prescribed displacements.

(4) ***imcma-unit-mappings.lisp:*** The events and signals for the new knowledge sources have been added to the list of existing events.

Appendix I. Tetrahedral Element Formulation - A Summary of Formulas

The finite element code for linear tetrahedral formulation has been written using references [10] - [12]. Reference [10] has been utilized for the thermal analysis formulation while reference [11] has been used to formulate the static analysis code. The general formula for the element thermal conductance matrix with coordinate axis aligned with orthotropic material conductivity tensor is

$$K_{ij}^e = \int_{\Omega^e} \left(K_{xx}^e \frac{\partial N_i^e}{\partial x}\frac{\partial N_j^e}{\partial x} + K_{yy}^e \frac{\partial N_i^e}{\partial y}\frac{\partial N_j^e}{\partial y} + K_{zz}^e \frac{\partial N_i^e}{\partial z}\frac{\partial N_i^e}{\partial z}] \, d\Omega \right) \quad i,j = 1,2,\ldots,N^e \qquad (75)$$

where Ω^e is the element volume, K^e_{xx}, K^e_{yy}, and K^e_{zz} are the element thermal conductivities in the x,y, and z directions, respectively, N_i^e's are the element nodal shape functions, and N^e is the number of nodes for element e. For the four-noded tetrahedral element, Eqn. (75) can be integrated in closed form to give

$$[K^e] = \frac{K_{xx}^e}{36\Omega^e}\begin{bmatrix} b_1b_1 & b_1b_2 & b_1b_3 & b_1b_4 \\ b_2b_1 & b_2b_2 & b_2b_3 & b_2b_4 \\ b_3b_1 & b_3b_2 & b_3b_3 & b_3b_4 \\ b_4b_1 & b_4b_2 & b_4b_3 & b_4b_4 \end{bmatrix} + \frac{K_{yy}^e}{36\Omega^e}\begin{bmatrix} c_1c_1 & c_1c_2 & c_1c_3 & c_1c_4 \\ c_2c_1 & c_2c_2 & c_2c_3 & c_2c_4 \\ c_3c_1 & c_3c_2 & c_3c_3 & c_3c_4 \\ c_4c_1 & c_4c_2 & c_4c_3 & c_4c_4 \end{bmatrix}$$
$$+\frac{K_{zz}}{36\Omega^e}\begin{bmatrix} d_1d_1 & d_1d_2 & d_1d_3 & d_1d_4 \\ d_2d_1 & d_2d_2 & d_2d_3 & d_2d_4 \\ d_3d_1 & d_3d_2 & d_3d_3 & d_3d_4 \\ d_4d_1 & d_4d_2 & d_4d_3 & d_4d_4 \end{bmatrix} \qquad (76)$$

where

$$b_1 = (y_2 - y_4)(z_3 - z_4) - (y_3 - y_4)(z_2 - z_4)$$
$$b_2 = (y_3 - y_4)(z_1 - z_4) - (y_1 - y_4)(z_3 - z_4)$$
$$b_3 = (y_1 - y_4)(z_2 - z_4) - (y_2 - y_4)(z_1 - z_4)$$
$$b_4 = -(b_1 + b_2 + b_3)$$

$$
\begin{aligned}
c_1 &= (x_3 - x_4)(z_2 - z_4) - (x_2 - x_4)(z_3 - z_4) \\
c_2 &= (x_1 - x_4)(z_3 - z_4) - (x_3 - x_4)(z_1 - z_4) \\
c_3 &= (x_2 - x_4)(z_1 - z_4) - (x_1 - x_4)(z_2 - z_4) \\
c_4 &= -(c_1 + c_2 + c_3) \\
d_1 &= (x_2 - x_4)(y_3 - y_4) - (x_3 - x_4)(y_2 - y_4) \\
d_2 &= (x_3 - x_4)(y_1 - y_4) - (x_1 - x_4)(y_3 - y_4) \\
d_3 &= (x_1 - x_4)(y_2 - y_4) - (x_2 - x_4)(y_1 - y_4) \\
d_4 &= -(d_1 + d_2 + d_3)
\end{aligned}
\tag{77}
$$

The elemental K matrix for a thermal analysis for a four noded tetrahedral element is a 4 X 4 matrix ($N^e \cdot NDF \times N^e \cdot NDF$) where *NDF* is the number of degrees of freedom per node which is unity for thermal problems. In case of a static analysis the elemental K matrix is a 12 X 12 matrix since there are three degrees of freedom per node corresponding to displacements in the x,y, and z direction.

For a thermal analysis the elemental load vector is as follows

$$\{r^e\} = \{r^e_{\bar{q}_B}\} + \{r^e_Q\} + \{r^e_{\bar{h}}\} \tag{78}$$

where $\{r^e_{\bar{q}_B}\}$ is the thermal load due to prescribed face flux $\bar{q}_B$, $\{r^e_Q\}$ is the load vector due to internal heat generation, and $\{r^e_{\bar{h}}\}$ is the thermal load due to convection boundary conditions on a face. The integral expressions and resulting values for these vectors are

$$\{r^e_{\bar{q}_B}\} = \int_{\Gamma^e} [N^e]^T \bar{q}^e_B \, d\Gamma = \frac{\bar{q}^e_B A_{123}}{3} \lfloor\ 1\ 1\ 1\ 0\ \rfloor^T \tag{79}$$

$$\{r^e_Q\} = \int_{\Omega^e} [N^e]^T Q^e \, d\Omega = \frac{Q^e \Omega^e}{4} \lfloor\ 1\ 1\ 1\ 1\ \rfloor^T \tag{80}$$

$$\{r^e_{\bar{h}}\} = \int_{\Gamma^e} [N^e]^T_i \bar{h}^e \, T^e_a \, d\Gamma = \frac{\bar{h}^e T^e_a A_{234}}{3} \lfloor\ 0\ 1\ 1\ 1\ \rfloor^T \tag{81}$$

where we have taken $\bar{q}^e_B, Q^e, \bar{h}^e$, and T^e_a to be constant over the element integrals, A_{123} and A_{234} are the surface areas of element faces defined by nodes 1-2-3 and 2-3-4, respectively, and we have assumed the flux to be applied on face 1-2-3 and convection boundary conditions acting on

face 2-3-4 for illustrative purposes. The above formulae show that all integrals are reduced to straightforward calculations and evaluations at nodal points.

In case of a static analysis, the strain vector for an element is defined as

$$\{\varepsilon^e\} = \{\ \varepsilon_x\ \ \varepsilon_y\ \ \varepsilon_z\ \ \gamma_{xy}\ \ \gamma_{yz}\ \ \gamma_{yz}\ \}^T \tag{82}$$

or in terms of the [B] matrix

$$\{\varepsilon^e\} = [B^e]\{d^e\} = \left[\ [B]_1\ \ [B]_2\ \ [B]_3\ \ [B]_4\ \right]\{d^e\} \tag{83}$$

where

$$[B^e]_i = \frac{1}{6\Omega^e}\begin{Bmatrix} b_i & 0 & 0 \\ 0 & c_i & 0 \\ 0 & 0 & d_i \\ c_i & b_i & 0 \\ 0 & d_i & c_i \\ d_i & 0 & b_i \end{Bmatrix} \quad i = 1,2,3,4 \tag{84}$$

The b, c and d's are same as previously defined and Ω^e is the volume of the element.

In general the stress vector in three-dimensional elasticity is given by

$$\{\sigma^e\} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{Bmatrix} = [E^e](\{\varepsilon^e\} - \{\varepsilon_0^e\}) + \{\sigma_o^e\} \tag{85}$$

where $[E^e]$ is the constitutive material matrix, $\{\varepsilon^e_0\}$ and $\{\sigma^e_0\}$ are the initial strain and stress acting on the element. Initial stresses may be induced by manufacturing processes as residual stresses in the material and are difficult to measure or predict. Initial strains are typically due to thermal expansion or contraction of the material due to heating or cooling of the material from its strain free reference temperature T_{ref}. Thermally induced initial strains are hydrostatic in nature

and are given by

$$\{\varepsilon_0^e\} = (T^e - T_{ref}^e)\begin{Bmatrix} \alpha_{xx}^e \\ \alpha_{yy}^e \\ \alpha_{zz}^e \\ 0 \\ 0 \\ 0 \end{Bmatrix} \tag{86}$$

where T^e is the element's temperature field and α^e_{xx}, α^e_{yy}, and α^e_{zz} are the elements coefficients of thermal expansion in the x,y, and z direction, respectively. The coupling of thermal behavior with stress analysis is now evident, since the temperature field within an element is required to compute initial strains. The initial strain vector produces a static load vector according to Eqn. (65) and affects the stress state according to Eqn. (85).

Appendix J. Sample Input File for FEECAP 2.5.

The following is the INPUT.DAT file for the new version of FEECAP. As can be noted from the analysis type index (ANAL TYP = 3), the input file is for a combined analysis. The lines marked '@1' are absent for a static analysis and those marked '@2' are absent for a thermal analysis.

The model is that of a cube the vertices of which are nodes. There is an additional node at the center of the cube. The model has nine nodes. Twelve, four noded tetrahedral elements are required to mesh this model. The elemental data in the input file shows the nodal connectivity for each element. All possible boundary conditions that may exist have been shown.

```
TITLE
TETRAHEDRAL ELEMENT INCORPORATION
NUMNP, NUMEG, MODEX, ANAL. TYP, IDTYP
9,    1,    1,    3, 3

NODAL DATA
1  1  1  1  0   0.0000000    1.000000    1.000000    2.000000
2  0  1  1  0   0.0000000   -1.000000    1.000000    2.000000
3  0  0  1  0   0.0000000   -1.000000   -1.000000    2.000000
4  0  0  1  1   40.000000    1.000000   -1.000000    2.000000
5  0  0  1  0   0.0000000    1.000000    1.000000    0.000000
6  0  0  0  1   80.000000   -1.000000    1.000000    0.000000
7  0  1  0  0   0.0000000   -1.000000   -1.000000    0.000000
8  0  0  0  1   90.000000    1.000000   -1.000000    0.000000
9  0  0  0  0    0.000000    0.000000    0.000000    1.000000

NELTYP, # ELEM, # MAT'LS,  NPE
3,    12,    1,    4
MAT'L NO., E,   NU,  MIN_TOL, MAX_TOL, BLOCK#
1,  0.540E+08, 0.2200, 0.1000, 0.2000, 1
TKr,      TKs ,  TKz,   BETA [DEGREES]                                    @1
10.00000,  10.00000,  10.00000,   0.000000                                @1
Alpha-R,  Alpha-S,  Alpha-Z   Tref                                        @1
0.640000007E-05,  0.640000007E-05,  0.640000007E-05,  0.0000000000        @1

NODAL CONNECTIVITY
```

```
1   1   5   4   9   1    0
2   8   4   5   9   1    0
3   7   4   8   9   1    100
4   7   3   4   9   1    0
5   2   4   3   9   1    0
6   2   1   4   9   1    0
7   2   3   7   9   1    100
8   2   7   6   9   1    0
9   2   5   1   9   1    0
10  2   6   5   9   1    0
11  7   8   6   9   1    100
12  6   8   5   9   1    0

FACE CONVECTION CARDS                                   @1
2                                                       @1
ELEM#   FACE#   H   TAMB                                @1
2   1   0.6  100                                        @1
4   4   0.8  100                                        @1

FACE FLUX CARDS                                         @1
2                                                       @1
ELEM#  FACE#  FLUX                                      @1
6   1    100                                            @1
3   2    100                                            @1

THERMAL LOAD DATA                                       @1
2                                                       @1
2  100                                                  @1
4  100                                                  @1

STATIC LOAD DATA                                        @2
4                                                       @2
5  3  -100                                              @2
6  3  -100                                              @2
7  3  -100                                              @2
8  3  -100                                              @2

PRES NODAL DISP                                         @2
2                                                       @2
5  3  1e-5                                              @2
7  2  2e-5                                              @2
```

*U.S. GOVERNMENT PRINTING OFFICE: 1995-610-126-50129

MISSION

OF

ROME LABORATORY

Mission. The mission of Rome Laboratory is to advance the science and technologies of command, control, communications and intelligence and to transition them into systems to meet customer needs. To achieve this, Rome Lab:

a. Conducts vigorous research, development and test programs in all applicable technologies;

b. Transitions technology to current and future systems to improve operational capability, readiness, and supportability;

c. Provides a full range of technical support to Air Force Materiel Command product centers and other Air Force organizations;

d. Promotes transfer of technology to the private sector;

e. Maintains leading edge technological expertise in the areas of surveillance, communications, command and control, intelligence, reliability science, electro-magnetic technology, photonics, signal processing, and computational science.

The thrust areas of technical competence include: Surveillance, Communications, Command and Control, Intelligence, Signal Processing, Computer Science and Technology, Electromagnetic Technology, Photonics and Reliability Sciences.